Check Point VSX

NGX R67

Administration Guide

20 February 2012

Check Point
SOFTWARE TECHNOLOGIES LTD.
We Secure the Internet.

Classification: [Protected]

© 2012 Check Point Software Technologies Ltd.

All rights reserved. This product and related documentation are protected by copyright and distributed under licensing restricting their use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form or by any means without prior written authorization of Check Point. While every precaution has been taken in the preparation of this book, Check Point assumes no responsibility for errors or omissions. This publication and features described herein are subject to change without notice.

RESTRICTED RIGHTS LEGEND:

Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

TRADEMARKS:

Refer to the Copyright page (http://www.checkpoint.com/copyright.html) for a list of our trademarks.

Refer to the Third Party copyright notices (http://www.checkpoint.com/3rd_party_copyright.html) for a list of relevant copyrights and third-party licenses.

Important Information

Latest Software

We recommend that you install the most recent software release to stay up-to-date with the latest functional improvements, stability fixes, security enhancements and protection against new and evolving attacks.

Latest Documentation

The latest version of this document is at:
http://supportcontent.checkpoint.com/documentation_download?ID=10165

For additional technical information, visit the Check Point Support Center (http://supportcenter.checkpoint.com).

Revision History

Date	Description
20 February 2012	Removed Hardware Monitoring using WebUI
2 May 2010	First release of this document

Feedback

Check Point is engaged in a continuous effort to improve its documentation.

Please help us by sending your comments (mailto:cp_techpub_feedback@checkpoint.com?subject=Feedback on Check Point VSX NGX R67 Administration Guide).

Contents

Important Information....3
Introduction to VSX 9
Product Names.... 9
VSX Glossary....10
VSX Overview....10
How VSX Works....11
Physical Network Topology....11
VSX Virtual Network Topology....12
Key Features and Benefits....12
Scalable Virtual Environment....12
High Performance Security....12
Non-Stop Security....13
Active/Standby Bridge Mode....13
Link Aggregation....13
SecurePlatform....13
URL Filtering....13
Hardware Health Monitoring....13
Typical VSX Deployments....13
VSX Architecture and Concepts 15
Overview....15
The VSX Gateway....15
Management Server Connections....15
Management Interface....17
Virtual Devices....18
Virtual System....18
Virtual System in Bridge Mode....18
Virtual Routers....19
Virtual Switches....20
Interfaces....20
VSX Management Overview....22
Introduction....22
Security Management Model....23
Multi-Domain Security Management Model....23
Management Model Comparison....24
Management Server Communication - SIC....24
VSX Traffic Flow....25
Overview....25
Context Determination....25
Security Enforcement....27
Forwarding to Destination....27
VSX Routing Concepts....27
Routing Overview....27
Routing Between Virtual Systems....27
Source-Based Routing....29
NAT....30
Dynamic Routing....30
VSX Clusters....30
High Availability....31
Virtual System Load Sharing (VSLS)....31
Configuring VSX 32
Overview....32
Rules & Security Policies....32

Working with VSX Gateways 32
Creating a New VSX Gateway 32
Modifying VSX Gateway Definitions 37
Deleting a VSX Gateway 41
VSX Gateway Recovery 42
Working with Virtual Systems 42
Creating a New Virtual System 42
Modifying a Virtual System Definition 47
Deleting a Virtual System 51
Working with Virtual Switches 51
Adding Virtual Switches 51
Modifying Virtual Switches 52
Deleting a Virtual Switch 53
Working with Virtual Routers 53
Creating a New Virtual Router 55
Modifying a Virtual Router Definition 56
Deleting a Virtual Router 58
Working with Source-Based Routing 58
Working with Dynamic Routing 59
Working with Interface Definitions 60
Adding a New Interface 60
Modifying an Interface Definition 63
Deleting an Interface 63
Working with Authentication 63
Supported Authentication Schemes 63
Configuring RADIUS or TACACS/TACACS+ 64
Configuring SecurID ACE/Server 64
Client/Session Authentication 66
VSX Limitations 66
Configuring Client/Session Authentication 66
Working with Network Address Translation 68
Configuring NAT 69
Tracking Activity with SmartView Monitor 69
Using VSX with Multi-Domain Security Management 70
Overview 70
Licensing VSX with Multi-Domain Security Management 71
Multi-Domain Security Management Licenses 71
VSX Gateway/Cluster Member Licenses 72
VSX/Domain Management Server Bundle Licenses 72
Upgrading Licenses 73
The Trial Period 73
License Violations 73
For More Information 74
VSX Provisioning 74
Defining Multi-Domain Servers 74
Installing a New Multi-Domain Server 74
Defining an Additional Multi-Domain Server in the SmartDomain Manager 75
Defining Domains and Servers 76
Creating a New Domain Object 76
Modifying Existing Domains and Servers 81
Working with Virtual Devices 81
Adding Virtual System to a Domain Management Server 81
Introduction to VSX Clusters 82
VSX Clustering Overview 82
Physical Clusters 82
VSX Clusters 83
Supported Cluster Environments 83
Planning a Cluster Deployment 83
VSX Cluster Architecture 84

VSX High Availability 85
VSX Gateway High Availability 85
Per Virtual System High Availability 85
Virtual System Load Sharing (VSLS) 86
Requirements 86
Conceptual Overview 86
Failure Recovery 90
Bridge Mode 90
Spanning Tree Protocol (STP) Bridge Mode 90
Active/Standby Bridge Mode 90
Using Virtual Switches in a Cluster 92
Managing VSX Clusters 93
Configuration Overview 93
Creating a New Cluster 93
Defining Cluster General Properties 94
Selecting Creation Templates 94
Adding Members 95
Defining Cluster Interfaces 96
Configuring Cluster Members 97
Cluster Management 97
Completing the Wizard 98
Modifying a Cluster Definition 98
Modifying Cluster Properties 98
Working with Cluster Members 107
Adding a New Member 107
Deleting a Member 108
Upgrading Cluster Members 108
Changing the Cluster Type 110
Converting from VSLS to High Availability 110
Converting from High Availability to VSLS 111
Sample Command Output 111
Configuring VSX High Availability 112
Enabling VSX Gateway High Availability 112
Enabling Per Virtual System High Availability 113
Configuring Virtual System Load Sharing 113
Enabling VSLS 113
Creating a New VSLS Cluster 114
Using the vsx_util vsls Command 114
Distributing Virtual Systems Amongst Members 115
Viewing VSLS Status 116
Exporting and Importing VSLS Configurations 117
Configuring Virtual Systems in Bridge Mode 119
Overview 119
STP Bridge Mode 119
Active/Standby Bridge Mode 121
Advanced Clustering Configuration 122
Clusters on the Same Layer-2 Segment 122
Monitoring all VLANs with ClusterXL 123
Enabling Dynamic Routing Protocols 124
Working with URL Filtering 126
Introduction 126
Terminology 126
Configuring URL Filtering 127
Enabling URL Filtering 127
Defining the URL Filtering Policy 127
Updating the Content Inspection Database 128
Password Bypass 129
URL Filtering Acceleration 129
Working with Link Aggregation 130

Link Aggregation Overview....130
Link Aggregation Terminology....130
How Link Aggregation Works....131
High Availability Overview....131
Load Sharing Overview....132
Bond Failover....132
Failover Support for VLANs....133
Bond Interface & Interface Limitations....133
Configuring Link Aggregation for High Availability....134
Creating a New Deployment....134
Upgrading an Existing Deployment....135
Link Aggregation - Load Sharing Mode....137
Creating a Bond in a New Deployment....138
Upgrading an Existing Deployment....141
Configuring Cisco Switches for Load Sharing....145
Changing the Bond Interface Mode....145
Enslaving Interfaces to a Bond....145
Detaching Interfaces from a Bond....146
Deleting a Bond....146
Removing a Bond Interface from Virtual devices....147
Removing a Bond Interface From a VSX Object....147
Removing a Bond Interface from a VSX Gateway or Cluster Member....147
Reconfiguring Interface Connections....147
Changing an Existing Interface to a Bond....148
Troubleshooting Bonded Interfaces....148
Troubleshooting Workflow....148
Optimizing VSX....150
VSX Resource Control....150
Overview....150
Resource Control System Components....150
Virtual System Priorities....151
Working with VSX Resource Control....151
QoS Enforcement....153
Overview....153
Architecture....154
QoS Features....155
QoS Management....155
QoS Configuration....156
Hardware Health Monitoring....161
Introduction to Hardware Health Monitoring....161
RAID Monitoring with SNMP....162
Example RAID Monitoring OIDs....163
Sensors Monitoring with SNMP on VSX-1 Appliances....163
Example Sensors Monitoring OIDs....164
Sensors Monitoring with SNMP on Power-1 and UTM-1 Appliances....164
Deploying VSX....166
Introduction....166
Internal Network Deployment Strategies....166
Security Gateway Deployment on a Physical Network....166
VSX Virtual System Deployment Strategies....167
Physical Internal Interface for Each Virtual System....167
Virtual Systems with Internal VLAN Interfaces....167
Internal Virtual Router with Source-Based Routing....168
Virtual Systems in the Bridge Mode....169
Cluster Deployments....169
Organizational Deployment Strategies....172
Enterprise Deployments....172
Managed Service Providers Using Multi-Domain Security Management....174
Data Centers....176

Migrating from an Open Server to a VSX-1 Appliance 177
VSX Diagnostics and Troubleshooting 179
Introduction 179
General Troubleshooting Steps 179
Troubleshooting Specific Problems 180
Cannot Establish SIC Trust for Gateway or Cluster 180
SIC Trust Problems with new Virtual Devices 180
Re-establishing SIC Trust with Virtual Devices 181
Sync Networks Do Not match 181
Install Policy Error Using VSX Creation Wizard 181
Internal Host Cannot Ping Virtual System 182
Command Line Reference 183
Firewall Commands 183
fw getifs 183
fw monitor 184
fw tab 184
fw fetch 185
VSX Command 187
vsx fetch 187
vsx fetchvs 188
vsx get 188
vsx set 188
vsx stat 189
vsx start_dr 190
vsx sic reset 191
Link Aggregation CLI Commands 191
cphaconf show_bond 191
chpaconf failover_bond 192
cphaprob -a if 192
VSX Resource Control Commands 192
vsx resctrl enforce 193
vsx resctrl monitor 193
vsx resctrl traffic_stat 193
vsx resctrl reset 193
vsx resctrl start 194
vsx resctrl stat 194
The vsx_util Command 195
add_member 196
add_member_reconf 197
change_interfaces 197
change_mgmt_ip 198
change_mgmt_private_net 198
fw fetch 199
change_interfaces 199
change_mgmt_subnet 201
convert_cluster 201
reconfigure 201
remove_member 202
show_interfaces 202
upgrade 203
view_vs_conf 203
vsls 205
The cphaprob Command 206
Index 209

Chapter 1

Introduction to VSX

In This Chapter

Product Names	9
VSX Glossary	10
VSX Overview	10
How VSX Works	11
Key Features and Benefits	12
Typical VSX Deployments	13

Product Names

Explanations and procedures included in this Administration Guide can apply to several brand names representing editions or variations of Check Point products. This document uses generic product names for variations of similar Check Point products.

The table below shows the generic product names used in this document and their product variations:

Generic Product Name	Includes the Following Products
Security Gateway	VPN-1 Power VPN-1 UTM VPN-1 UTM Edge VPN-1 UTM Embedded VPN-1 Pro VPN-1 Express Any other Check Point products with VPN-1 functionality
Multi-Domain Security Management	Multi-Domain Security Management SiteManager-1
SecurePlatform	SecurePlatform SecurePlatform Pro

VSX Glossary

Term	Definition
VSX	**V**irtual **S**ystem E**x**tension - Check Point virtual networking solution, hosted on a single computer or cluster containing virtual abstractions of Check Point Security Gateways and other network devices. These virtual devices provide the same functionality as their physical counterparts.
VSX Gateway	Physical server that hosts VSX virtual networks, including all **virtual devices** that provide the functionality of physical network devices.
Management Server	The Security Gateway or a Multi-Domain Security Management Domain Management Server used by administrators to manage the VSX virtual network and and its security policies.
Virtual Device	Generic term for any VSX virtual network component
Virtual System	Virtual device that provides the functionality of a physical Security Gateway that provides full firewall VPN, and IPS functionality.
Virtual System in the Bridge Mode.	A Virtual System that implements native layer-2 bridging instead of IP routing, thereby enabling deployment of Virtual Systems in an existing topology without reconfiguring the IP routing scheme
Virtual Switch	Virtual device that provides the functionality of a physical switch in a VSX deployment
Virtual Router	Virtual device that provides the functionality of a physical router in a VSX deployment
Virtual Interface	Virtual device that provides the functionality of a physical interface on a virtual device
Warp (wrp) Link	A Virtual Interface that is created automatically in a VSX topology

VSX Overview

VSX (Virtual System Extension) is a security and VPN solution for large-scale environments based on the proven security of Check Point Security Gateway. VSX provides comprehensive protection for multiple networks or VLANs within complex infrastructures. It securely connects them to shared resources such as the Internet and/or a DMZ, and allows them to safely interact with each other. VSX is supported by IPS™ Services, which provide up-to-date preemptive security.

VSX incorporates the same patented Stateful Inspection and Application Intelligence technologies used in the Check Point Security Gateway product line. It runs on high speed platforms (known as **VSX gateways**) to deliver superior performance in high-bandwidth environments. Administrators manage VSX using a Security Gateway or a Multi-Domain Security Management Multi-Domain Server, delivering a unified management architecture that supports enterprises and service providers.

A VSX gateway contains a complete set of virtual devices that function as physical network components, such as Security Gateway, routers, switches, interfaces, and even network cables. Centrally managed, and incorporating key network resources internally, VSX allows businesses to deploy comprehensive firewall and VPN functionality, while reducing hardware investment and improving efficiency.

How VSX Works

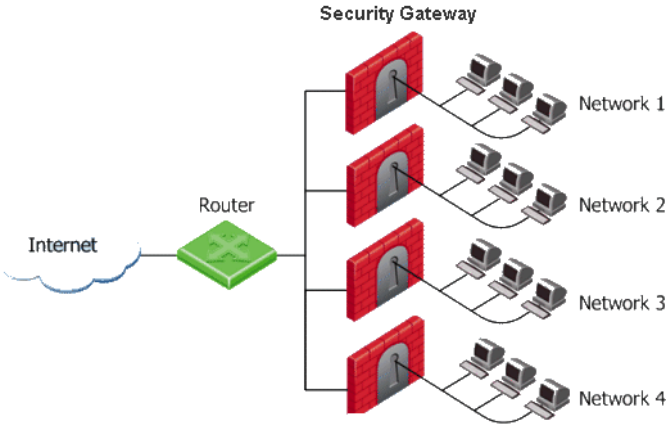
Each "virtual" Security Gateway (known as a **Virtual System** in VSX terminology) functions as an independent firewall, protecting a specific network. Once packets arrive at the VSX gateway, it directs traffic to the Virtual System protecting the destination network. The Virtual System inspects all traffic and passes or rejects it according to rules contained in its Rule Base.

In order to better understand how virtual networks work, it is important to compare physical network environments with their virtual (VSX) counterparts. While physical networks consist of many hardware components, VSX virtual networks reside on a single configurable VSX gateway or cluster that defines and protects multiple independent networks, together with their virtual components.

Physical Network Topology

The figure below shows a typical deployment with four physical Security Gateways, each protecting a separate network. Each Security Gateway is a separate, physical machine that is hard-wired to the perimeter router and its corresponding network.

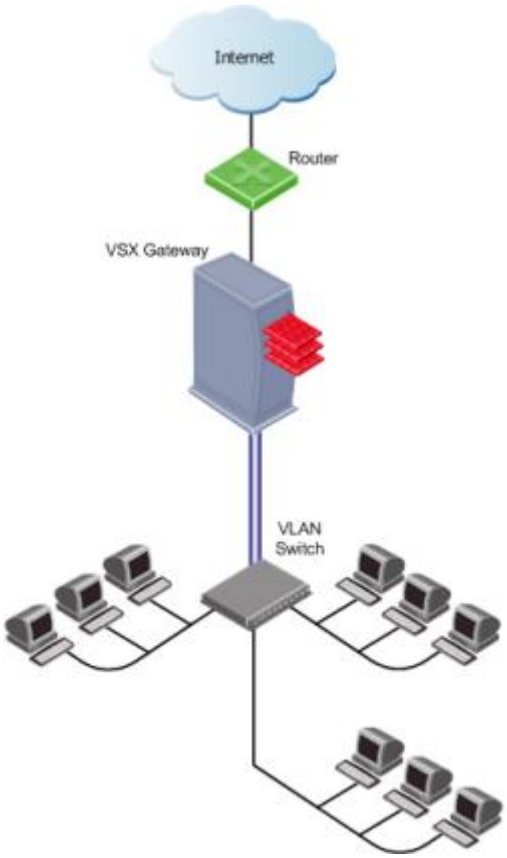
Figure 1-1 Separate physical gateways protecting each network



VSX Virtual Network Topology

The example shows how a single VSX gateway, in this case containing four Virtual Systems, protects all four networks.

Figure 1-2 A VSX gateway replaces multiple physical gateways



Each Virtual System in a VSX environment works as an individual Security Gateway, providing the same security and networking functionality as a physical gateway. This example also shows:

- Four Virtual Systems, each handling packet traffic to and from discrete networks.
- One Virtual Switch providing connectivity for all the Virtual Systems to the Internet router.
- "Virtual" interfaces and network cables (known as **Warp Links**) providing point-to-point connections between the Virtual Systems and the Virtual Switch.

Key Features and Benefits

Scalable Virtual Environment

Up to 250 virtual devices can be deployed on a single VSX gateway or VSX cluster, providing a highly scalable virtual platform while reducing hardware investment, space requirements, and maintenance costs.

High Performance Security

High-bandwidth networks require high-performance gateways in order to support thousands of applications and users. To provide security at wire speed, VSX can be deployed on multiple carrier-class platforms using Check Point's SecureXL™ performance technology, ensuring secure, multi-gigabit throughput.

Virtual System Load Sharing (VSLS) provides the ability to distribute Virtual Systems across cluster members, effectively distributing Virtual System traffic load within a cluster.

VSX Resource Control allows administrators to manage the processing load by guaranteeing that each Virtual System will receive its minimum CPU allocation. Resources not needed by one Virtual System are automatically made available to other Virtual Systems.

VSX QoS Enforcement provides the ability to control network quality of service in the VSX network environment by supporting the Differentiated Services (DiffServe) protocol and assigning different transmission characteristics to different classes of service.

Non-Stop Security

VSX supports the Check Point ClusterXL technology as well as third-party cluster solutions, such as Crossbeam, to guarantee nonstop security. Seamless connection failover promotes high availability and resiliency, ensuring, nonstop, secure business operations at both the application and network levels.

Active/Standby Bridge Mode

The **Active/Standby Bridge Mode** enhances network resiliency by enabling instantaneous failover and by providing full support for VSLS in the Bridge Mode. This feature also provides full control over bridge failover.

Link Aggregation

Link Aggregation, also known as Interface Bonding, lets you join interfaces for High Availability or Load Sharing. This networking technology binds together multiple physical interfaces to increase reliability and throughput.

In a High Availability deployment, only one interface is active at a time. If that interface or connection fails, the bond manages the failover to a standby slave interface.

In a load sharing deployment, Link Aggregation significantly increases total throughput by spreading the traffic load amongst multiple interfaces. All interfaces are active, and traffic is balanced between interfaces. Load Sharing operates according to the IEEE 802.3ad or the XOR standard.

SecurePlatform

This release includes the latest enhancements to the SecurePlatform operating system.

SecurePlatform of this release is based on Linux kernel 2.6.18-92cp and Red Hat Enterprise Linux 5.2 for user mode components and supports a large variety of hardware, including open servers, network cards and RAID controllers. A comprehensive list of certified hardware can be found at: (http://www.checkpoint.com/products/supported_platforms/secureplatform.html)

URL Filtering

URL Filtering enforces filtering rules based on organizational needs and predefined categories made up of URLs and URL patterns. URL Filtering takes place according to predefined categories made up of URLs and/or IPs. The URL Filter checks the URL and/or IP of a Web page against a list of approved sites. In this way, complete sites or pages within sites that contain objectionable material (pornography, pirated music or videos, illegal software, etc.) can be blocked. In addition, the URL Filtering policy only checks connections that have already passed the security policy.

Hardware Health Monitoring

SecurePlatform includes new Hardware Health Monitoring capabilities, support for RAID and Sensors monitoring over SNMP.

Typical VSX Deployments

VSX virtual networking provides an ideal solution for a variety of deployment scenarios ("Deploying VSX" on page 166):

- Enterprises enforcing distinct security policies per department
- Internet service providers offering secure environments

- College campuses with many discrete networks for students, faculty and administration
- Any other large organization requiring multiple firewalls

In each case, VSX provides access control, NAT, VPN, remote access, logging, and IPS services.

Chapter 2

VSX Architecture and Concepts

In This Chapter

Overview	15
The VSX Gateway	15
Virtual Devices	18
VSX Management Overview	22
VSX Traffic Flow	25
VSX Routing Concepts	27
VSX Clusters	30

Overview

This chapter presents an overview of core VSX concepts and describes the architecture and building blocks that comprise a VSX virtual environment. This information is essential in order to plan, provision, configure, and operate a VSX virtual network deployment. VSX includes a robust set of virtual components that emulate the functionality of physical network devices. By using these virtual components, you can create network topologies that are functionally equivalent to physical networks.

The term "Virtual Devices" refers to Virtual Systems, Virtual Switches, and Virtual Routers.

This chapter also introduces the two principal management models with which you manage the VSX environment. Finally, this chapter describes several routing and traffic management features that are applicable to VSX environments.

The VSX Gateway

A VSX gateway is a physical machine that hosts virtual "networks", consisting of **virtual devices** that provide the functionality of their physical network counterparts such as: Security Gateways, routers and switches.

A VSX gateway performs the following tasks:

- Communicates with the management server to handle provisioning and configuration for all virtual devices
- Manages state synchronization to for high availability and for load sharing in cluster deployments.

Management Server Connections

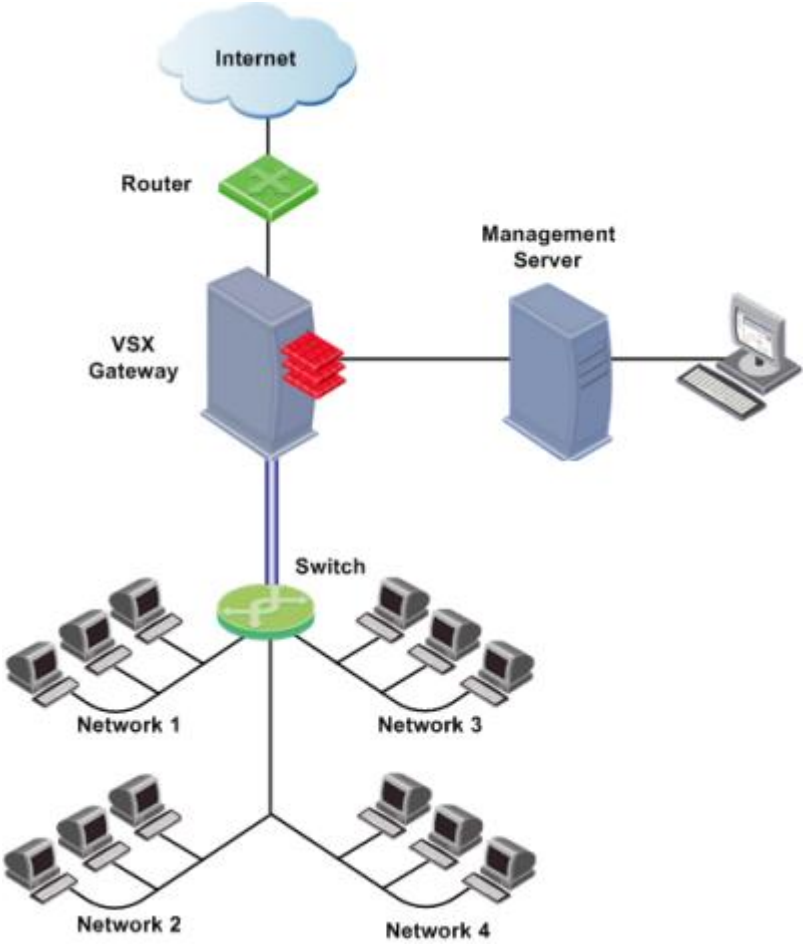
A management server (Security Gateway or Multi-Domain Security Management Multi-Domain Server) connects to the VSX gateway and provides provisioning and configuration services for virtual devices located on the VSX gateway. You can connect the management server to the VSX gateway using one of the following scenarios.

- **Local Connection**: The management server connects directly to the VSX gateway via a dedicated management interface.
- **Remote Connection**: The management server connects remotely from an external or internal network by means of a router connected to a management interface. This method ensures segregation of management traffic from all other traffic.

Local Management Connection

When using a local management server (Security Management Server or Multi-Domain Security Management), all management traffic is handled by a dedicated management interface (DMI) that connects the management server with the VSX gateway. The dedicated management interface IP address can be either private or public.

Figure 2-3 Typical VSX topology using local management

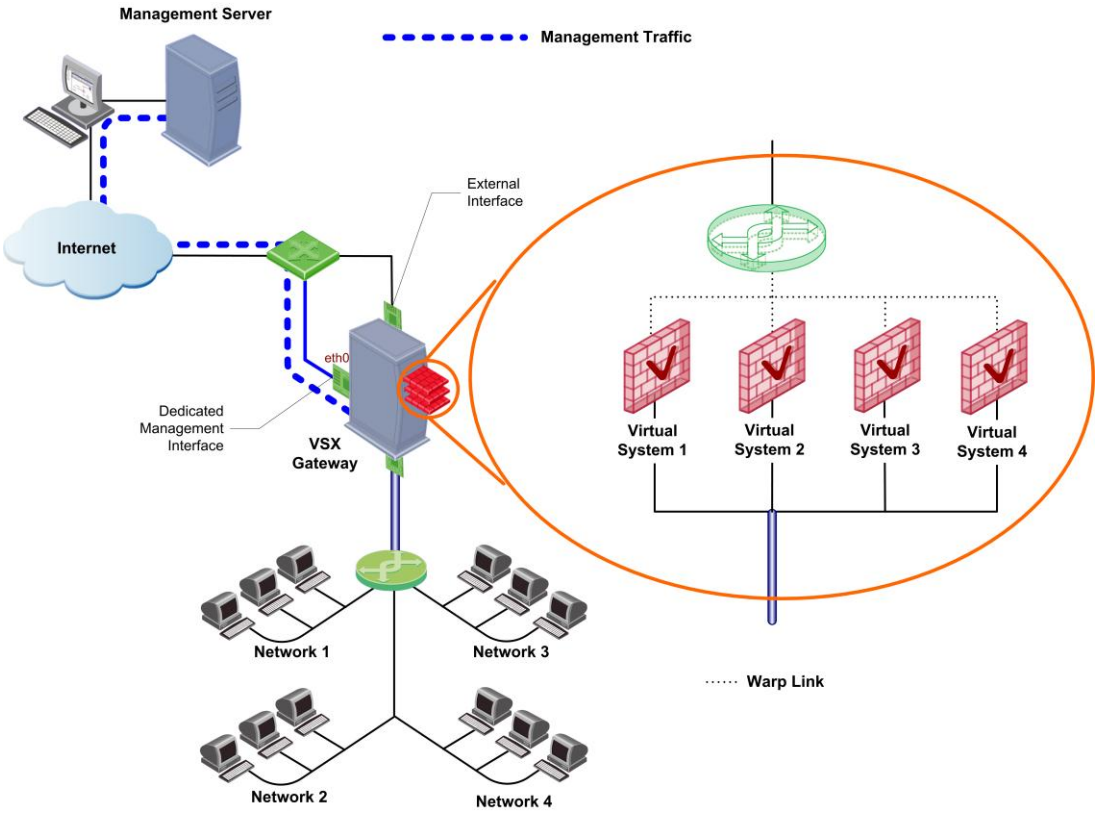


Remote Management connection

When using a remote management server (Security Gateway or Multi-Domain Security Management), management traffic travels via an internal or external network to a VSX gateway to the management interface. This architecture segregates management traffic from all other traffic passing through the VSX gateway.

Check Point recommends that remote management connections use a dedicated management interface (DMI) that connects directly to a router or switch that leads to the external network or the Internet. The following diagram illustrates this scenario.

Figure 2-4 Typical VSX deployment with DMI remote management



You can choose to use a non-dedicated management interface by connecting a Virtual Router or Virtual Switch to the management interface. This, however, is not recommended.

When management traffic passes through a Virtual Router or Switch, you must ensure that the associated Warp Link IP address originates from the remote network. Furthermore, if the remote management connection arrives via the Internet, you must assign a routable, public IP address.

Management Interface

A VSX deployment can be managed using one of the following interface schemes:

- **Dedicated Management Interface (DMI)**: Uses a separate interface that is restricted to management traffic, such as provisioning, logging and monitoring
- **Non-Dedicated Management Interface**: Uses a shared internal or external interface that also carries routine user traffic

Dedicated Management Interface (DMI)

Check Point recommends that you use a DMI for management for the following reasons:

- Segregation of management traffic from routine "production" traffic enhance performance, especially for end users
- Enables several advanced VSX features

Non-Dedicated Management Interface

VSX supports non-DMI deployments primarily to provide backward compatibility with legacy deployments. When configuring a non-DMI deployment, you can define remote management connections only via a Virtual Switch or Virtual Router. Remote management connects via a Virtual System are not supported.

Check Point does not recommend using non-DMI for the following reasons:

- Provisioning and logging may degrade user performance
- Does not support several new VSX features
- Non-DMI is irreversible - you cannot change a non-DMI gateway to DMI

Virtual Devices

This section describes virtual network components and their characteristics.

Virtual System

A Virtual System is a virtual security and routing domain that provides the functionality of a Security Gateway with full firewall and VPN facilities. Multiple Virtual Systems can run concurrently on a single VSX gateway.

Virtual System Autonomy

Each virtual system functions as a stand-alone, independent entity, much in the same way as each Security Gateway is independent from other gateways. Each Virtual System maintains its own interfaces, IP addresses, routing table, ARP table and dynamic routing configuration. In addition, each Virtual System maintains its own:

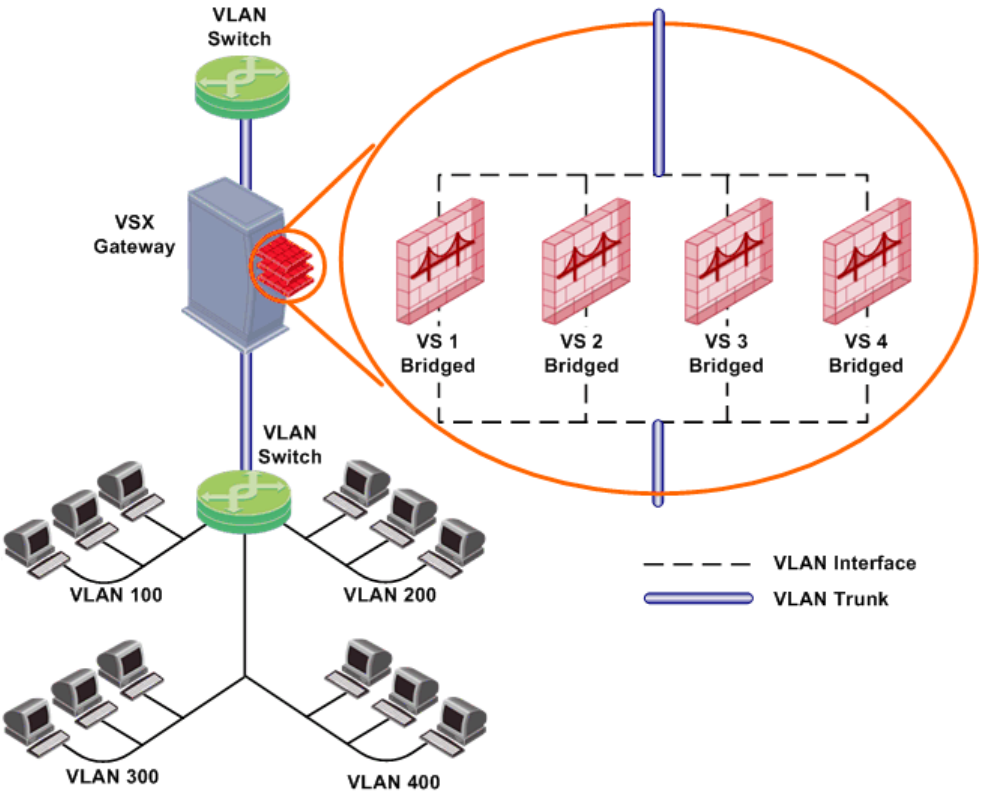
- **State Tables**: Each Virtual System contains its own kernel tables containing configuration and runtime data, such as, active connections, IPSec tunnel information, etc.
- **Security and VPN policies**: Each Virtual System enforces its own security and VPN Policies (including INSPECT code). Policies are retrieved from the management server and stored separately on the local disk and in the kernel. In a Multi-Domain Security Management environment, each Domain database is maintained separately on the management server as well as on the VSX gateway.
- **Configuration Parameters**: Each Virtual System maintains its own configuration, such as IPS settings, TCP/UDP time-outs, etc.
- **Logging Configuration**: Each Virtual System maintains its own logs and performs logging according to its own rules and configuration.

Virtual System in Bridge Mode

A Virtual System in the bridge mode implements native layer-2 bridging instead of IP routing. This allows network administrators to easily and transparently deploy a Virtual System in an existing network topology without reconfiguring the existing IP routing scheme.

A typical bridge mode scenario incorporates an 802.1q compatible VLAN switch on either side of the VSX gateway. The Virtual System interfaces do not require IP addresses and it remains transparent to the existing IP network.

Figure 2-5 Virtual System in the Bridge Mode



A Virtual System in the bridge mode:

- Has the same security capabilities as a Virtual System, except for VPN and NAT
- Simplifies virtual network management
- Does not segment an existing virtual network
- Requires manual topology configuration in order to enforce anti-spoofing

Virtual Routers

A **Virtual Router** is an independent routing domain within a VSX gateway that performs the functionality of physical routers. Virtual Routers are useful for connecting multiple Virtual Systems to a shared interface, such as the interface leading to the Internet, and for routing traffic from one Virtual System to another. Virtual Routers support dynamic routing.

Virtual Routers perform the following routing functions:

- Packets arriving at the VSX gateway through a shared interface to the designated Virtual System based on the source or destination IP address.
- Traffic arriving from Virtual Systems directed to a shared interface or to other Virtual Systems.
- Traffic to and from shared network resources such as a DMZ.

As with physical routers, each Virtual Router maintains a routing table with a list of route entries describing known networks and directions on how to reach them. Depending on the deployment requirements, multiple Virtual Routers can be configured.

To protect themselves, Virtual Routers inspect all traffic destined to, or emanating from themselves (for example, an ICMP ping to the Virtual Router IP address) based on the security policy. Traffic that is not destined to, or emanating from the Virtual Router is not inspected by the Virtual Router policy and is forwarded to its destination.

Virtual Switches

By providing layer-2 connectivity, a **Virtual Switch** connects Virtual Systems and facilitates sharing a common physical interface without segmenting the existing IP network. As with a physical switch, each Virtual Switch maintains a forwarding table with a list of MAC addresses and their associated ports.

In contrast to a Virtual Router, when sharing a physical interface via a Virtual Switch there is no need:

- To allocate an additional subnet for IP addresses of Virtual Systems connected to the switch.
- To manually configure the routing on the routers adjacent to the shared interface.

You can create multiple Virtual Switches in a virtual network topology.

Note - When sharing a physical interface via a Virtual Switch, the IP addresses for Virtual Systems connected to a Virtual Switch should be allocated from the same subnet as the shared interface.

If the only function the Virtual Switch performs is to connect Virtual Systems, then the Virtual Switch can be defined without interfaces (unless Virtual System load sharing is enabled).

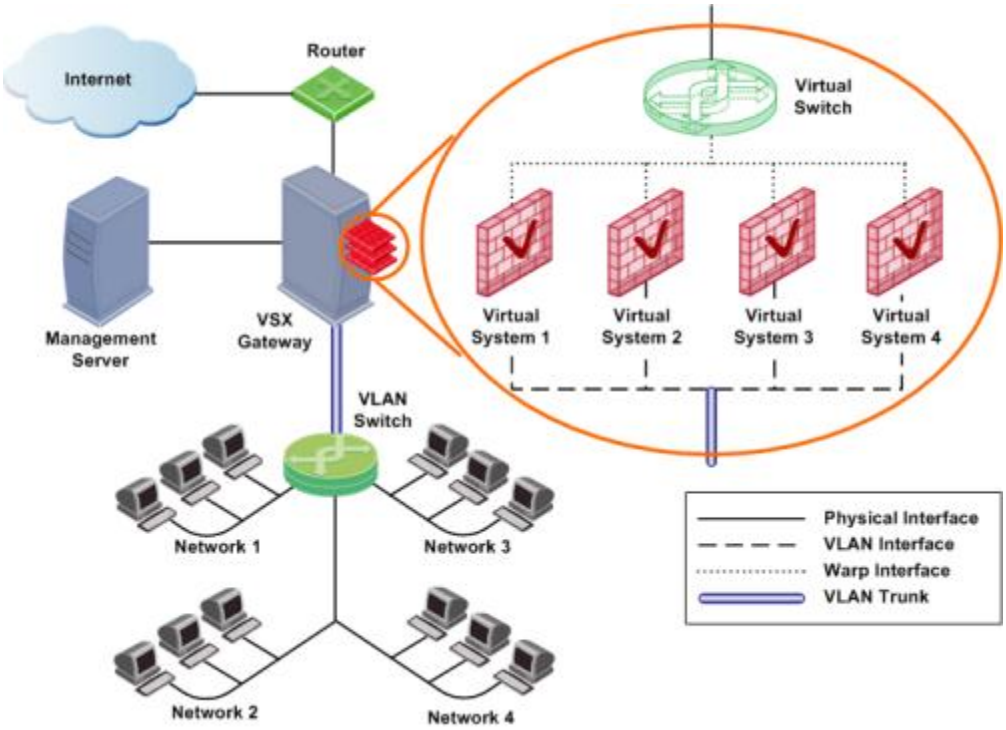
Interfaces

This section describes the various types of interfaces and how they are used in a VSX configuration. The principal interface types are:

- Physical Interface
- VLAN interface
- Warp Link (including unnumbered interfaces)

The following figure presents a simple example that illustrates how the various interface types are used in a VSX environment.

Figure 2-6 VSX interface types



In the above figure:

- Warp Links connect the Virtual Switch to each Virtual System.

- A Physical Interface connects the Virtual Switch to an external router leading to the Internet.
- VLAN Interfaces connect the Virtual Systems to the VLAN Switch, via A VLAN trunk.
- The VLAN switch connects to the protected networks.

Physical Interfaces

Physical interfaces connect a VSX gateway to internal and external networks, as well as to the management server. There are three types of physical interfaces (four types for a VSX Cluster) used in a VSX gateway:

- **Dedicated Management Interface**: Connects the VSX gateway to the management server when it is locally managed. If the VSX gateway is remotely managed, then the management connection arrives via the external or internal interface.
- **External interface**: Connects the VSX gateway to the Internet or other untrusted networks.
- **Internal Interface**: Connects the VSX gateway to a protected network.
- **Synchronization Interface**: Connects one VSX gateway member to other members for state synchronization in a VSX clustering deployment.

Additional physical interfaces can be installed and attached to any virtual device as required. A VSX gateway can theoretically contain as many physical interfaces as permitted by gateway hardware and memory constraints.

VLAN Interfaces

Virtual Systems typically connect to protected VLAN networks using IEEE 802.1q compliant VLAN Interfaces. The networks are connected to ports on an 802.1q-compliant switch that trunks all traffic via a single physical interface to the VSX gateway.

VSX uses VLAN tags to direct the Ethernet frames to the specific Virtual System handling each network. VSX assigns a virtual VLAN interface to each VLAN tag on a specific physical interface. For Example: VLAN tag 100 on eth3 will be assigned a virtual interface named eth3.100.

Warp Links

A Warp Link is a virtual point-to-point connection between a Virtual System and a Virtual Router or Virtual Switch. Each side of a Warp Link represents is a virtual interface with the appropriate virtual device.

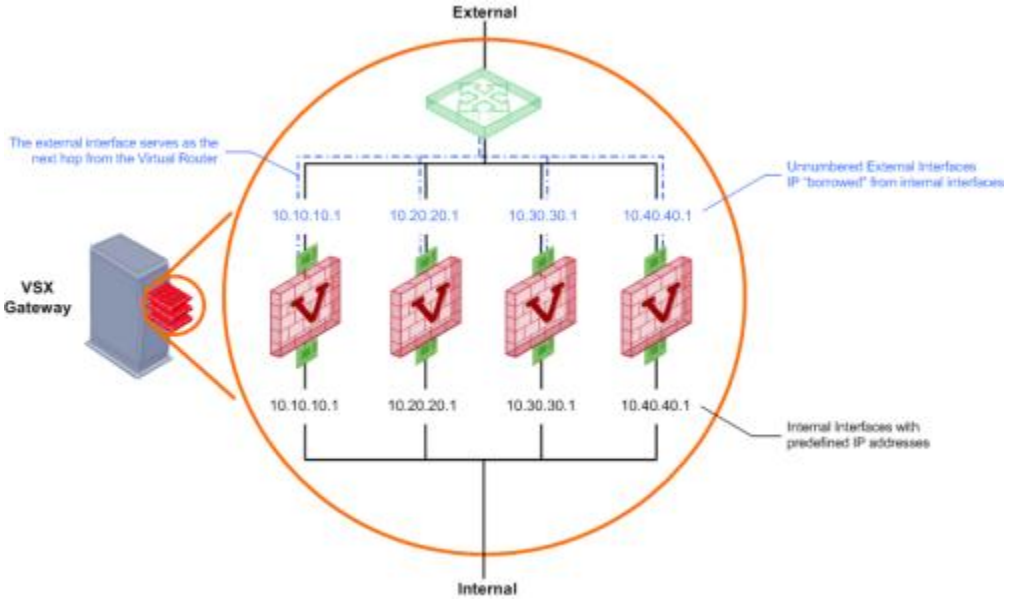
NGX R67 VSX automatically assigns a name to each virtual interface when administrators create the link. Warp Interfaces on the Virtual System side are assigned the prefix `wrp` and those on the Virtual Router/Switch side are assigned the prefix `wrpj`. In both cases, VSX appends a unique number to the prefix to form the interface name.

When connected to a Virtual Switch, VSX also assigns a unique MAC address to each Warp Link.

Unnumbered Interfaces

VSX allows you reduce the number of IP addresses required for a VSX network deployment when using one or more Virtual Routers. A Warp link connected to a Virtual Router can "borrow" an existing IP address from another interface, instead of assigning a dedicated address to the interface leading to a Virtual Router. This capability is known as an **Unnumbered Interface**.

Figure 2-7 Unnumbered interfaces



The above figure illustrates a topology using unnumbered interfaces. In this example, the external interfaces for each Virtual System are unnumbered and borrow the IP address of the internal interfaces. Unnumbered interfaces act as the next hop from the Virtual Router.

Unnumbered Interface Limitations

The following limitations apply to Unnumbered Interfaces:

- Unnumbered interfaces must connect to a Virtual Router.
- You can only "borrow" an individual interface IP address once.
- In order to use VPN or Hide NAT, the borrowed address must be routable.

VSX Management Overview

Introduction

VSX supports two Check Point management models: **Security Management** and Multi-Domain Security Management. Both models provide central configuration, management and monitoring for multiple VSX gateways and Virtual Systems. The choice of management model depends on several factors, including:

- The scale of the current deployment and anticipated expansion
- Administrative requirements
- Physical and operational requirements
- Licensing restrictions

You can use either management model to manage "physical" Security Gateway together with VSX gateways and Virtual Systems. You can also manage VPN communities and remote connections with either model.

Note - According to the Check Point EULA (End User License Agreement), a Security Gateway can only manage security policies for Virtual Systems belonging to a single legal entity. In order to manage Virtual Systems belonging to multiple legal entities, you need to deploy a Multi-Domain Security Management management solution with a separate Domain Management Server for each legal entity. For more information regarding Licensing, refer to your Check Point Reseller.

Security Management Model

The Security Management model is appropriate for enterprise deployments containing up to 25 Virtual Systems. In this model, SmartDashboard connects to the Security Gateway, which in turn manages the VSX gateway.'

The Security Gateway provides a single management domain with one object database to manage Virtual Devices as well as other *physical* devices. Only one administrator at a time can use SmartDashboard to provision Virtual Systems, and configure security policies.

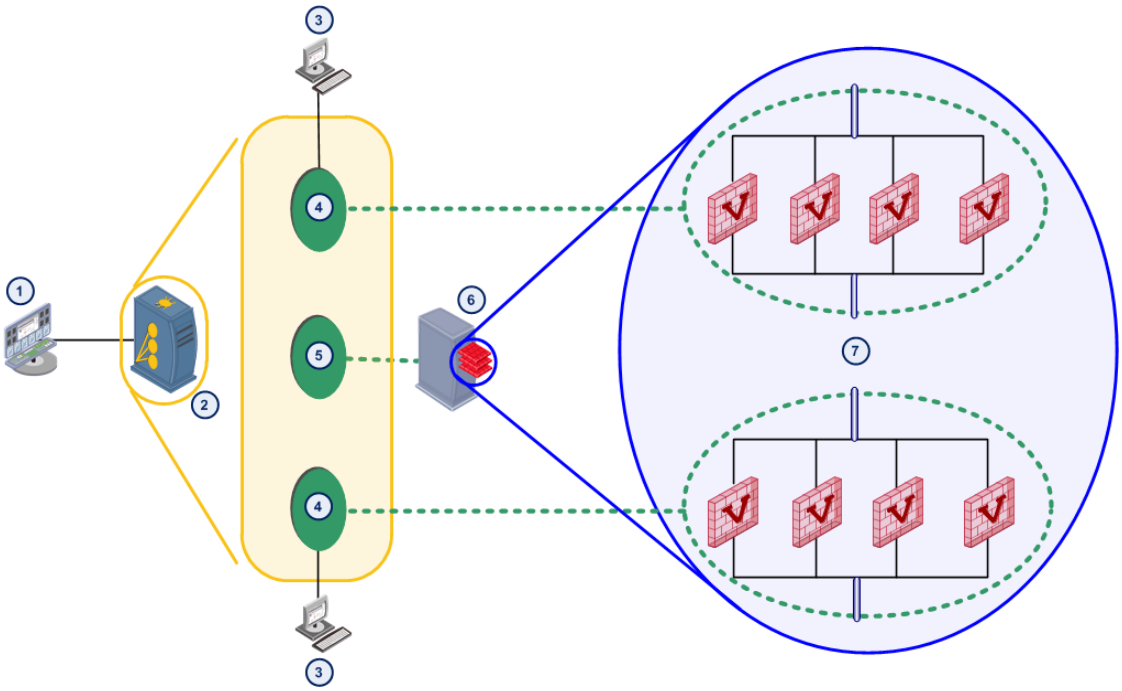
Multi-Domain Security Management Model

Using the Multi-Domain Security Management model, administrators centrally manage multiple independent networks, typically belonging to different Domains, divisions or branches. The **Multi-Domain Server** is the central management node that controls the network and security policy databases for each of these networks.

Each Domain network is managed by a **Domain Management Server**, which provides the full functionality of a Security Gateway and can host multiple Virtual Systems, virtual devices and physical devices. The server that manages the VSX gateway is the **Main Domain Management Server**.

Check Point recommends that each VSX gateway in a Multi-Domain Security Management deployment be managed by its own, separate, Main Domain Management Server. A VSX gateway can host Virtual Systems that are managed by different Domain Management Servers.

Figure 2-8 Multi-Domain Security Management Managing VSX



	Description

	Description
1	SmartDomain Manager
2	Multi-Domain Server
3	SmartDashboard
4	Domain Management Server
5	Main Domain Management Server
6	VSX Gateway
7	VSX Virtual System in Domain Management Servers

Using the **SmartDomain Manager**, you provision and configure Domains and Domain Management Servers. Each Domain Management Server uses its own SmartDashboard instance to provision and configure its Virtual Systems, virtual devices, and security policies.

Management Model Comparison

The following table summarizes the capabilities and differences between the two management models. The capacity figures shown for Multi-Domain Security Management represent estimated, practical limits that will sustain acceptable performance levels under normal conditions. Actual capacities and performance are a dependent on many factors, including deployed hardware, network topology, traffic load and security requirements.

Table 2-1 ***VSX Management Model Comparison***

Feature	Security Management Server	Multi-Domain Security Management (Practical Limit)
Management Domains	1	250
Concurrent Administrators	1	250
Object Databases	1	250
Policies	250	250
Certificate Authorities	1	250
Virtual Systems	25 (recommended)	250

Management Server Communication - SIC

All communication between the management server and the VSX gateway is accomplished by means of Secure Internal Communication (SIC), a certificate based channel that authenticates communication between Check Point components. The management server uses SIC for provisioning virtual devices, policy installation, logging, and status monitoring.

SIC trust is initially established using a one-time password during configuration of the VSX gateway or cluster members. For Multi-Domain Security Management deployments, SIC trust is established between the Domain Management Server associated with the VSX gateway or cluster (Main Domain Management Server).

Virtual devices establish trust in a different manner than their physical counterparts. When creating a virtual device, VSX automatically establishes SIC trust using the secure communication channel defined between the management server and the VSX gateway. The VSX gateway uses its management interface for Secure Internal Communication between the management server and all virtual devices.

VSX Traffic Flow

Overview

A VSX gateway processes traffic according to the following steps:

- Context determination
- Security enforcement
- Forwarding to destination

Context Determination

VSX incorporates **VRF** (Virtual Routing and Forwarding) technology that allows creation of multiple, independent routing domains on a single VSX gateway or cluster. The independence of these routing domains makes possible the use of virtual devices with overlapping IP addresses. Each routing domain is known as a **context**.

When traffic arrives at a VSX gateway, a process known as **Context Determination** directs traffic to the appropriate Virtual System, Virtual Router or Virtual Switch. The context determination process depends on the virtual network topology and the connectivity of the virtual devices.

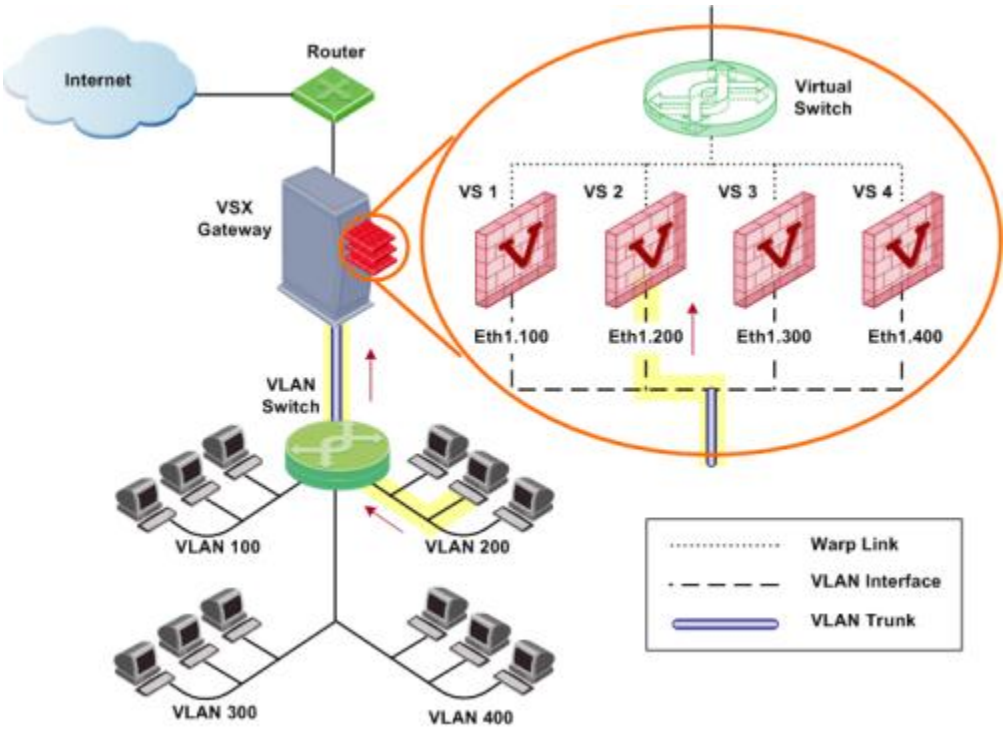
The three basic Virtual System connection scenarios are:

- Virtual System directly connected to a physical or VLAN interface
- Virtual System connected via a Virtual Switch
- Virtual System connected via a Virtual Router

Direct Connection to a Physical Interface

When traffic arrives at an interface (either physical or VLAN) that directly connects to a Virtual System, the connection itself determines the context and traffic passes directly to the appropriate Virtual System via that interface. In the following example, VSX automatically directs traffic arriving via VLAN Interface `eth1.200` to Virtual System 2 according to the context defined by the VLAN ID.

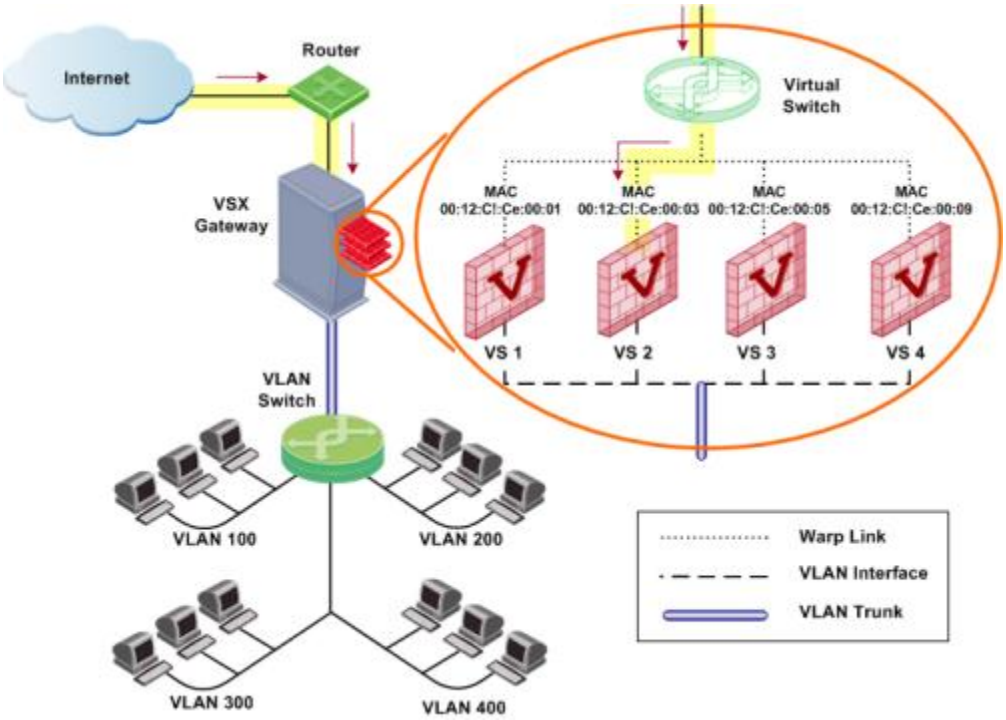
Figure 2-9 Directly connected interface example



Connection via a Virtual Switch

Traffic arriving via a Virtual Switch passes to the appropriate Virtual System based on the destination MAC address, as defined in the Virtual Switch forwarding table. Traffic arrives at the Virtual System via the Warp Link associated with the designated MAC address.

Figure 2-10 Typical Virtual Switch scenario

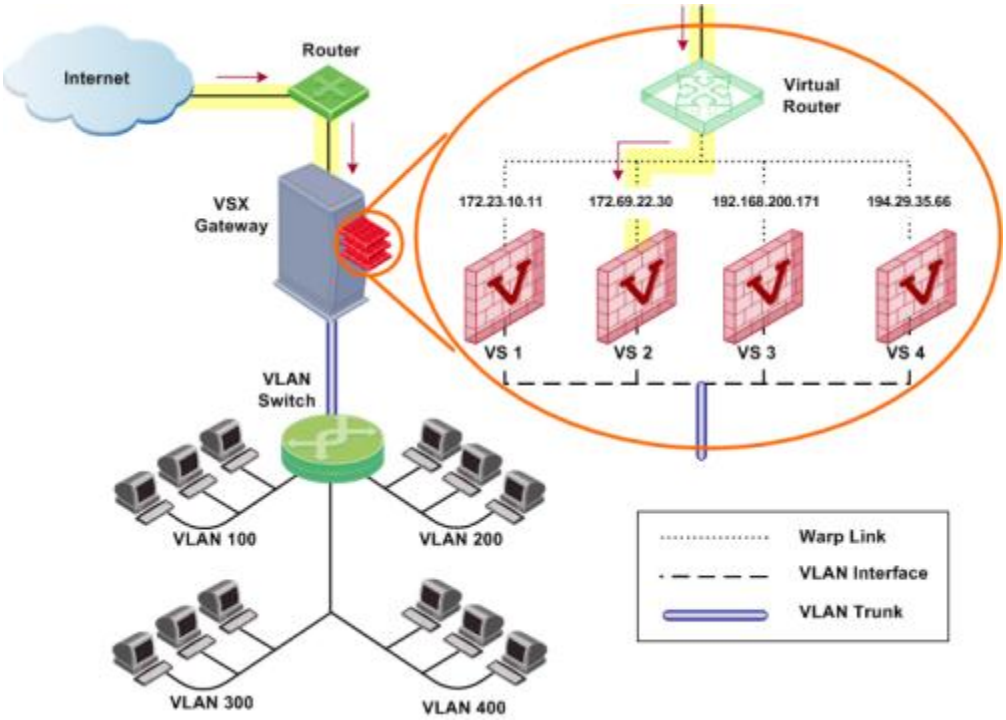


If the destination MAC address does not exist in the Virtual Switch forwarding table, the traffic is broadcast over all defined Warp Links. The Virtual Switch scenario is common for inbound traffic from external networks or the Internet.

Connection via a Virtual Router

Traffic arriving via a Virtual Router passes to the appropriate Virtual System based on entries in the Virtual Router routing table. Routing may be destination-based, source-based or both. Traffic arrives to the designated Virtual System via its warp link.

Figure 2-11 Typical Virtual Router Scenario



Security Enforcement

Since each Virtual System functions as an independent Security Gateway, it maintains its own, unique security policy to protect the network behind it. The designated Virtual System inspects all traffic and allows or blocks it based the rules contained in the security policy.

Forwarding to Destination

Each virtual system maintains its own unique configuration and rules for processing and forwarding traffic to its final destination. This configuration also includes definitions and rules for NAT, VPN, and other advanced features.

VSX Routing Concepts

Routing Overview

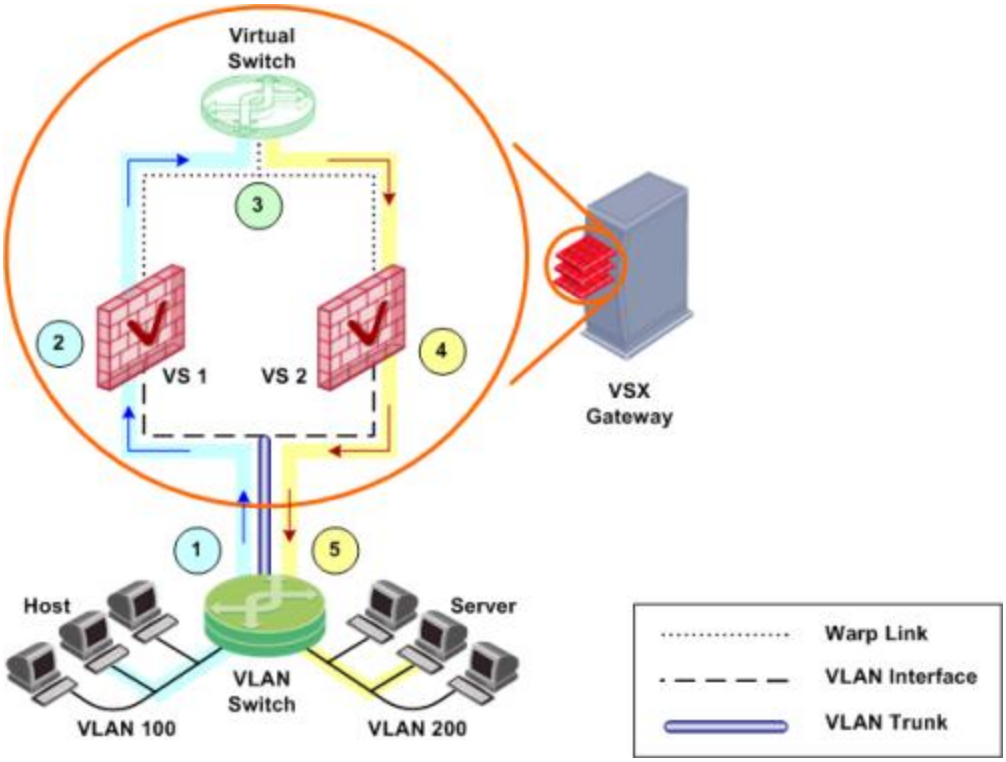
The traffic routing features in VSX network topologies are analogous to those available for physical networks. This section discusses several routing features and strategies as they apply to a VSX environment.

Routing Between Virtual Systems

Virtual Routers and Switches can be used to forward traffic between networks located behind virtual systems, much in the same manner as their physical counterparts.

The figure below presents an example of how Virtual Systems connected to a Virtual Switch and a physical VLAN switch communicate with each other. In this example, a host in VLAN 100 sends data to a server located in VLAN 200.

Figure 2-12 Routing of virtual traffic between Virtual Systems



1. Traffic from the VLAN 100 host arrives at the VLAN switch, which inserts a VLAN tag and passes it to the VSX gateway via a VLAN trunk.
2. Based on its VLAN tag, the VSX gateway assigns the traffic to the Virtual System named VS1. VS1 inspects the traffic according to its security policy and forwards the traffic on to the Virtual Switch.
3. VS1 "knows" to forward the traffic to VS2 via the Virtual Switch based on its routing configuration.
4. VS2 inspects the traffic according to its security policy, inserts a VLAN tag, and passes it to back the VLAN switch.
5. The VLAN switch forwards the traffic to the server located on VLAN 200.

Route Propagation

When a Virtual System is connected to a Virtual Router or to a Virtual Switch, you can choose to **propagate** its routing information to adjacent Virtual Devices. This feature enables network nodes located behind neighboring Virtual Systems to communicate without the need for manual configuration.

Route propagation works by automatically updating virtual device routing tables with routes leading to the appropriate Virtual Systems.

Route Propagation using a Virtual Router

When Virtual Systems are connected to a Virtual Router, VSX propagates routes by automatically adding entries to the routing table contained in the Virtual Router. Each entry contains a route pointing to the destination subnet using the Virtual System router-side Warp Interface (`wrpj`) as the next hop.

Route Propagation using a Virtual Switch

When Virtual Systems are connected to a Virtual Switch, VSX propagates routes by automatically adding entries to the routing table in each Virtual System. Each entry contains a route pointing to the destination subnet using the Virtual System Warp Interface (`wrp`) IP address.

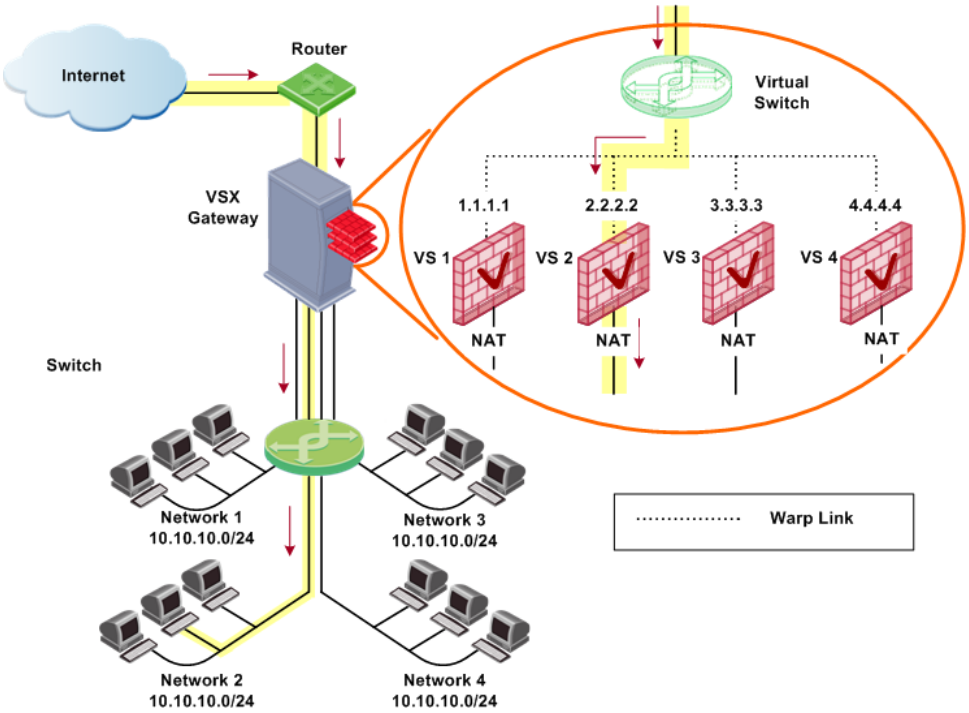
Overlapping IP Address Space

VSX facilitates connectivity when multiple network segments share the same IP address range (**IP address space**). This scenario occurs when a single VSX gateway protects several independent networks that assign IP addresses to endpoints from the same pool of IP addresses. Thus, it is feasible that more than one endpoint in a VSX environment will have the identical IP address, provided that each is located behind different Virtual System.

Overlapping IP address space in VSX environments is possible because each Virtual System maintains its own unique state and routing tables. These tables can contain identical entries, but within different, segregated contexts. Virtual Systems use NAT to facilitate mapping internal IP addresses to one or more external IP addresses.

The below figure demonstrates how traffic passes from the Internet to an internal network with overlapping IP address ranges, using NAT at each Virtual System.

Figure 2-13 Example of overlapping IP addresses



In this case, Network 1, Network 2 Network 3, and Network 4 all share the same network address pool, which might result in identical overlapping IP addresses. However, packets originating from or targeted to these networks are processed by their respective Virtual System using NAT to translate the original/overlapping addresses to unique routable addresses.

Additional Considerations for Virtual Switch Route Propagation

To update the topology map for each Virtual System, you still need to edit and save each Virtual System object that is connected to the Virtual Switch after enabling route propagation. You do not, however, need to manually define the topology, as this is done automatically.

Following the topology update, you must then re-install the security policy for the affected Virtual Systems. This procedure is necessary in order to ensure that the Anti-Spoofing and VPN features work properly.

Source-Based Routing

Source-based routing allows you to define routing definitions that take precedence over ordinary, destination-based, routing decisions. This allows you to route packets according to their source IP address or a combination of their source IP address and destination IP address.

Source-based routing is useful in deployments where a single physical interface without VLAN tagging connects several protected Domain networks. Each Virtual System is connected to an internal Virtual Router. The Virtual Router routes traffic to the appropriate Virtual System based on the source IP address, as defined in source-based routing rules.

Limitations

- Source-based routing does not support overlapping IP addresses.
- Anti-spoofing protection is not effective for packets origination form a shared internal interface because there is no physical or logical segregation of traffic. In this case, it is recommended that you deploy anti-spoofing protection on the router itself.

NAT

Virtual Systems support Network Address Translation (NAT), much in the same manner as a physical firewall. When a Virtual System, using either Static or Hide NAT, connects to a Virtual Router, you must propagate the affected routes to the virtual router. To do so, you need to first define NAT addresses for Virtual Systems connected to a Virtual Router.

The NAT configuration section ("Virtual System - NAT" on page 49) presents the configuration procedure for NAT on Virtual Machines.

Dynamic Routing

Virtual Devices can communicate and distribute routes amongst themselves using dynamic routing. VSX provides full layer-3 dynamic routing for Virtual Systems and Virtual Routers. The following unicast and multicast dynamic routing protocols are supported:

- OSPF
- RIP-v2
- BGP-v4
- IGMP
- PIM-SM
- PIM-DM

Dynamic routing is configured and stored separately for each Virtual Device. Each Virtual Devices has its own dynamic routing daemon.

VSX Clusters

A VSX cluster consists of two or more identical, interconnected VSX gateways that ensure continuous data synchronization and transparent failover. Furthermore, Virtual System Load Sharing (VSLS) enhances throughput by distributing Virtual Systems, together with their traffic load, amongst multiple, redundant machines.

VSX supports the following cluster environments:

- Check Point ClusterXL
- Crossbeam X-Series Chassis

VSX supports the following **Bridge Mode** solutions for ClusterXL deployments:

- **STP Bridge Mode**: Provides path redundancy while preventing undesirable loops between redundant switches.
- **Active/Standby Bridge Mode**: Provides full path redundancy and loop prevention, while offering seamless support for Virtual System Load Sharing and overcomes many STP limitations.

The VSX Clusters chapter ("Introduction to VSX Clusters" on page 82) provides detailed conceptual information, while the Cluster Management chapter ("Managing VSX Clusters" on page 93) provides detailed configuration procedures, including instructions for enabling and using all VSX clustering features. For more about Check Point ClusterXL features and functionality see the *R75.20 ClusterXL Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12265).

High Availability

VSX provides for high system availability by ensuring transparent failover for VSX gateways and/or for individual Virtual Systems. If the active VSX gateway member fails, all sessions continue to run, securely and without interruption, on a standby cluster member. If an individual Virtual System fails, you can configure that Virtual System to fail over to a standby member while all other Virtual Systems continue to function on the active VSX gateway member.

Users need not reconnect and re-authenticate, nor do they notice that an alternate machine has taken over. The Selective Sync feature allows you to selectively activate, delay or disable cluster member synchronization.

Virtual System Load Sharing (VSLS)

Load Sharing offers significant performance advantages while providing failover for individual Virtual Systems. Using multiple gateways instead of a single gateway significantly increases linear performance for CPU intensive applications such as VPNs, Security servers, Policy servers, and Active Directory (LDAP).

By distributing Virtual System instances between different cluster members, the performance load is efficiently spread amongst the members. For example, active Virtual System 1 runs on member A, while active Virtual System 2 runs on member B. Standby and backup Virtual system instances are likewise distributed amongst members to maximize throughput, even in a failover scenario.

VSLS provides an excellent scalability solution, allowing administrators to add additional physical members to an existing VSLS cluster as traffic loads and performance requirements increase.

VSLS is available only in a Check Point ClusterXL environment.

Chapter 3

Configuring VSX

In This Chapter

Overview	32
Working with VSX Gateways	32
Working with Virtual Systems	42
Working with Virtual Switches	51
Working with Virtual Routers	53
Working with Interface Definitions	60
Working with Authentication	63
Client/Session Authentication	66
Working with Network Address Translation	68
Tracking Activity with SmartView Monitor	69

Overview

This chapter explains how to provision, configure and manage a VSX virtual network environment. You use the SmartDashboard to provision and configure Virtual systems and other virtual devices.

If you define or configure VSX objects in a Multi-Domain Security Management deployment: open the SmartDashboard of the Domain Management Server that manages the virtual devices. The Multi-Domain Security Management chapter ("Using VSX with Multi-Domain Security Management" on page 70) explains these procedures.

To do the procedures explained in this chapter, the VSX gateway and the management servers (Security Management Server or Multi-Domain Server) must be running. You should have already installed the GUI clients (SmartDashboard or SmartDomain Manager) on the appropriate machines.

This chapter assumes that you are familiar with SmartDashboard and how to define standard Security Gateway objects and security policies. Many virtual device and policy operations are equivalent to those for physical Security Gateways. Therefore, these procedures are not presented in this Administration Guide.

Rules & Security Policies

Defining and installing security policies on a VSX gateway or Virtual System is the same as on a physical Security Gateway. Therefore, these procedures are not presented in this Administration Guide.

Working with VSX Gateways

A VSX gateway is a physical machine that serves as a container for Virtual Systems and other virtual network components. This section has step-by-step procedures for creating and configuring standalone VSX gateways.

Creating a New VSX Gateway

This section explains how to create a new VSX gateway using the **VSX Gateway Wizard**. After you complete the VSX Gateway Wizard, you can change the VSX gateway definition from SmartDashboard. For example, you can add or delete interfaces, or configure existing interfaces to support VLANs.

To use the VSX Gateway wizard**:**

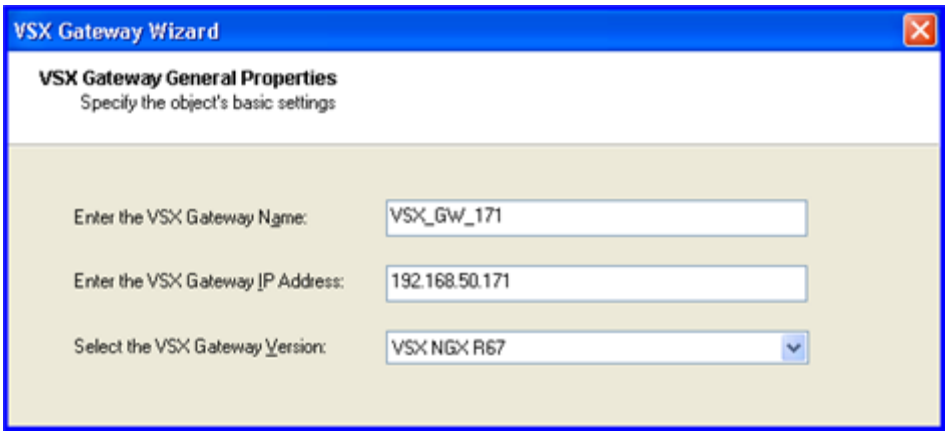
1. Open SmartDashboard.

   If you are using Multi-Domain Security Management, open SmartDashboard from the Domain Management Server of the VSX gateway.
2. In the **Network Objects** tab in the **Objects Tree**, right-click **Check Point** and select **New Check Point**.
3. Select the VSX type and then select **Gateway**.

   The **VSX Gateway Wizard** opens, showing the **General Properties** page.

Defining VSX Gateway General Properties

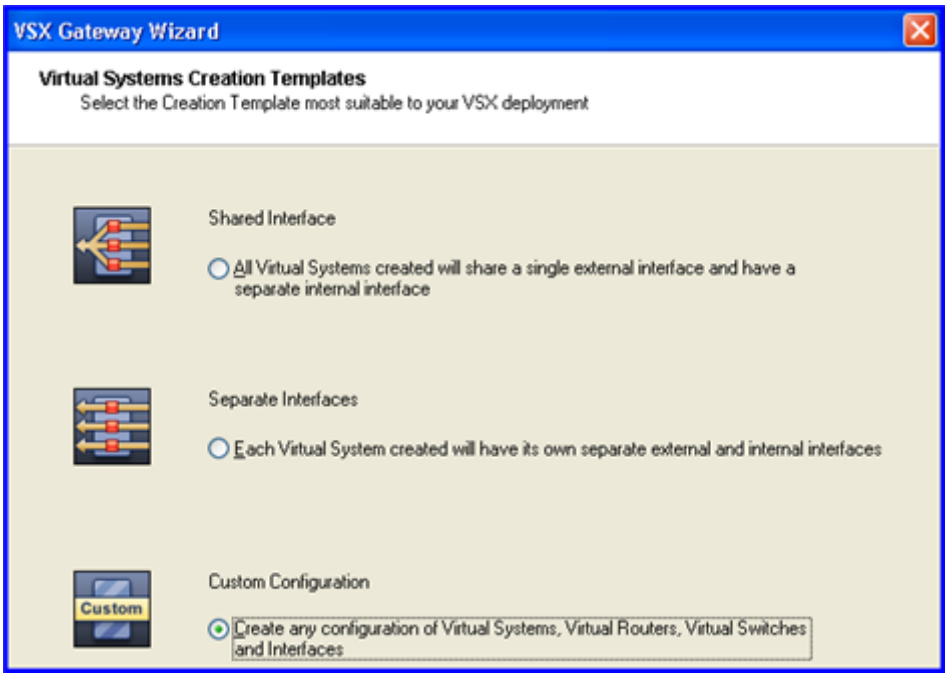
The **General Properties** page contains basic identification properties for VSX gateways.



- **VSX Gateway Name**: Unique, alphanumeric for the VSX gateway. The name cannot contain spaces or special characters except the underscore.
- **VSX Gateway IP Address**: Management interface IP address.
- **VSX Gateway Version**: Select the VSX version installed on the VSX gateway from the drop-down list.

Selecting Creation Templates

The **Creation Templates** page lets you provision predefined, default topology and routing definitions to Virtual Systems. This makes sure Virtual Systems are consistent and makes the definition process faster. You always have the option to override the default creation template when you create or change a Virtual System.


VSX Gateway Wizard
Virtual Systems Creation Templates
Select the Creation Template most suitable to your VSX deployment
Shared Interface
All Virtual Systems created will share a single external interface and have a separate internal interface
Separate Interfaces
Each Virtual System created will have its own separate external and internal interfaces
Custom Configuration
Custom
Create any configuration of Virtual Systems, Virtual Routers, Virtual Switches and Interfaces

The default Creation Templates are:

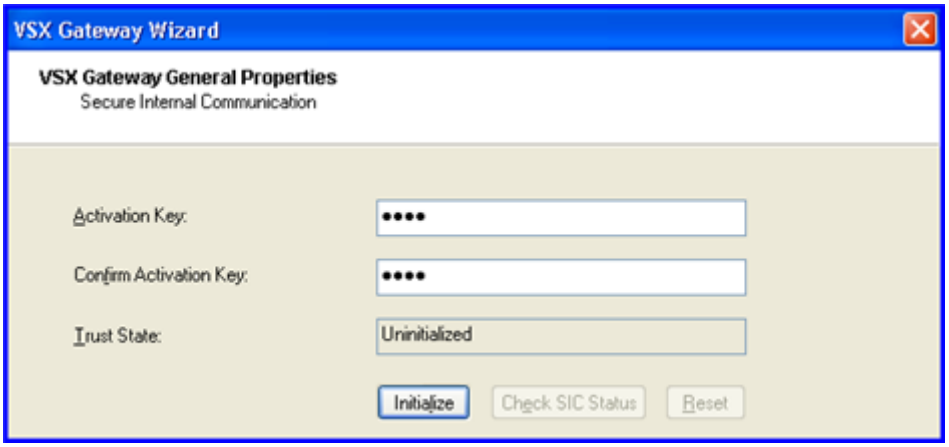
- **Shared Interface**: Virtual systems share one external interface, but maintain separate internal interfaces.
- **Separate Interfaces**: Virtual systems use their own separate internal and external interfaces. This template creates a Dedicated Management Interface (DMI) by default.

If the default templates are not appropriate, you can create a custom configuration:

- **Custom Configuration**: Define Virtual System, Virtual Router, Virtual Switch, and Interface configurations.

Establishing SIC Trust

Initialize Secure Internal Communication trust between the VSX gateway and the management server. The gateway and server cannot communicate without Trust.



Initializing SIC Trust

When you create a VSX gateway, you must give an Activation Key. Enter and confirm the activation key and then click **Initialize**. If you enter the correct activation key, the **Trust State** changes to `Trust established`.

Troubleshooting SIC Trust Initialization Problems

If SIC trust was not successfully established, click **Check SIC Status** to see the reason for the failure. The most common issues are an incorrect activation key and connectivity problems between the management server and the VSX gateway.

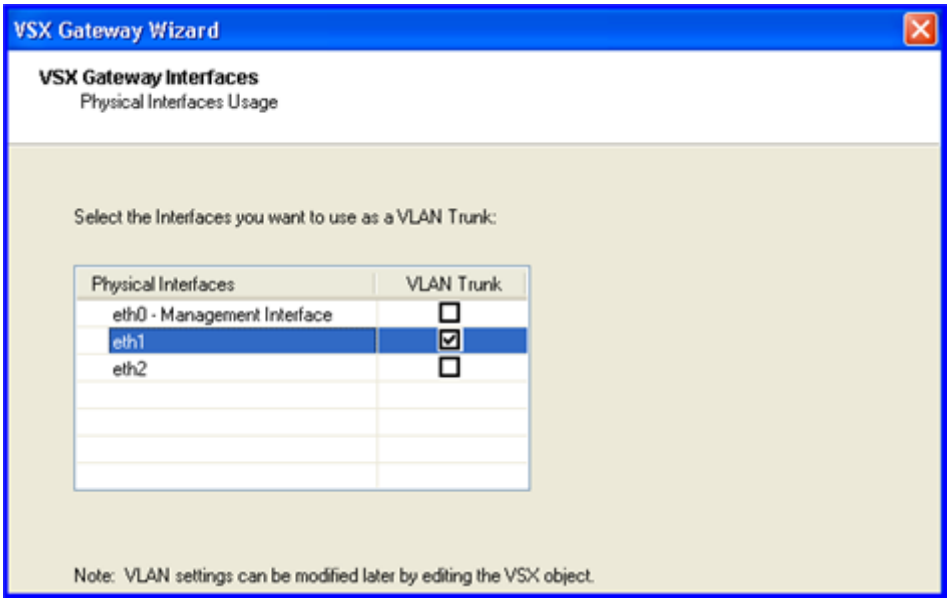
Troubleshooting to resolve SIC initialization problems:

- Re-enter and re-confirm the activation key.
- Verify that the IP address defined in **General Properties** is correct.
- Ping the management server to verify connectivity. Resolve connectivity issues.
- From the VSX gateway command line, use the `cpconfig` utility to re-initialize SIC. After this process completes, click **Reset** in the wizard and then re-enter the activation key.

See the *R75 Security Management Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11667).

Defining Physical Interfaces

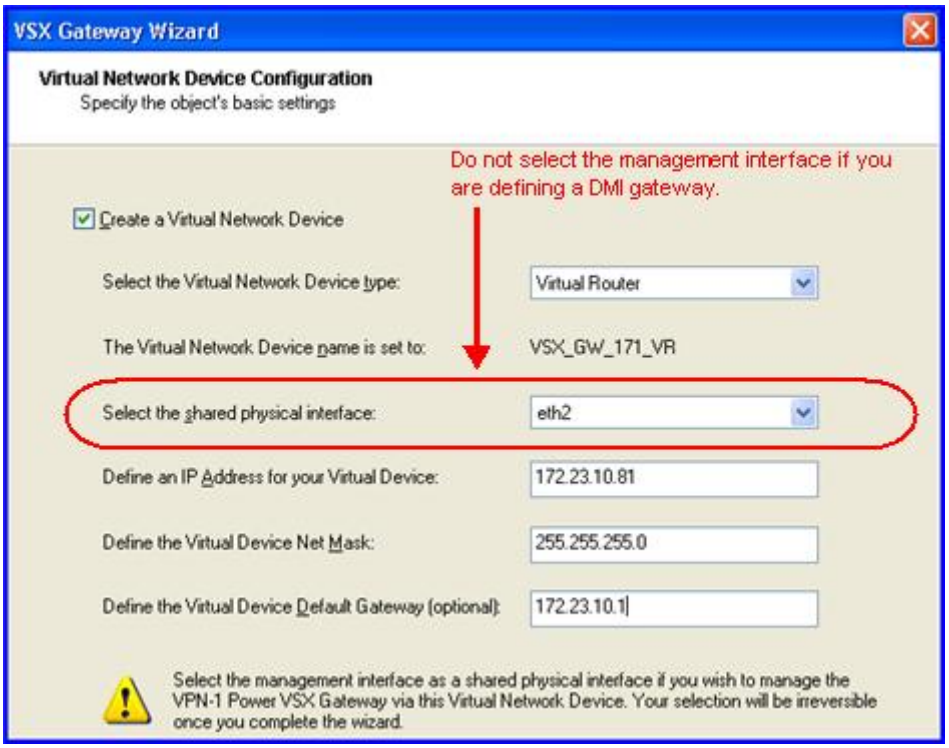
In the **VSX Gateway Interfaces** window, define physical interfaces as VLAN trunks. The table shows the interfaces currently defined on the gateway machine.


VSX Gateway Wizard
VSX Gateway Interfaces
Physical Interfaces Usage
Select the Interfaces you want to use as a VLAN Trunk:
Physical Interfaces
VLAN Trunk
eth0 - Management Interface
eth1
eth2
Note: VLAN settings can be modified later by editing the VSX object.

To define an interface as a VLAN trunk, select **VLAN Trunk**.

Virtual Network Device Configuration

If you chose the **Custom Configuration** option, the **Virtual Network Device Configuration** window opens. In this window, define a Virtual Device with an interface shared with the VSX gateway. If you do not want to define a Virtual Device at this time, click **Next** to continue.


VSX Gateway Wizard
Virtual Network Device Configuration
Specify the object's basic settings
Do not select the management interface if you are defining a DMI gateway.
Create a Virtual Network Device
Select the Virtual Network Device type:
Virtual Router
The Virtual Network Device name is set to:
VSX_GW_171_VR
Select the shared physical interface:
eth2
Define an IP Address for your Virtual Device:
172.23.10.81
Define the Virtual Device Net Mask:
255.255.255.0
Define the Virtual Device Default Gateway (optional):
172.23.10.1
Select the management interface as a shared physical interface if you wish to manage the VPN-1 Power VSX Gateway via this Virtual Network Device. Your selection will be irreversible once you complete the wizard.

To define a virtual device with a shared interface:

1. Select **Create a Virtual Device**.
2. Select the **Virtual Network Device type** (**Virtual Router** or **Virtual Switch**).
3. Select the **shared physical interface** to define a non-DMI gateway.

   Do not select the management interface if you want to define a **Dedicated Management Interface (DMI)** gateway. If you do not define a shared Virtual Device, a DMI gateway is created by default.

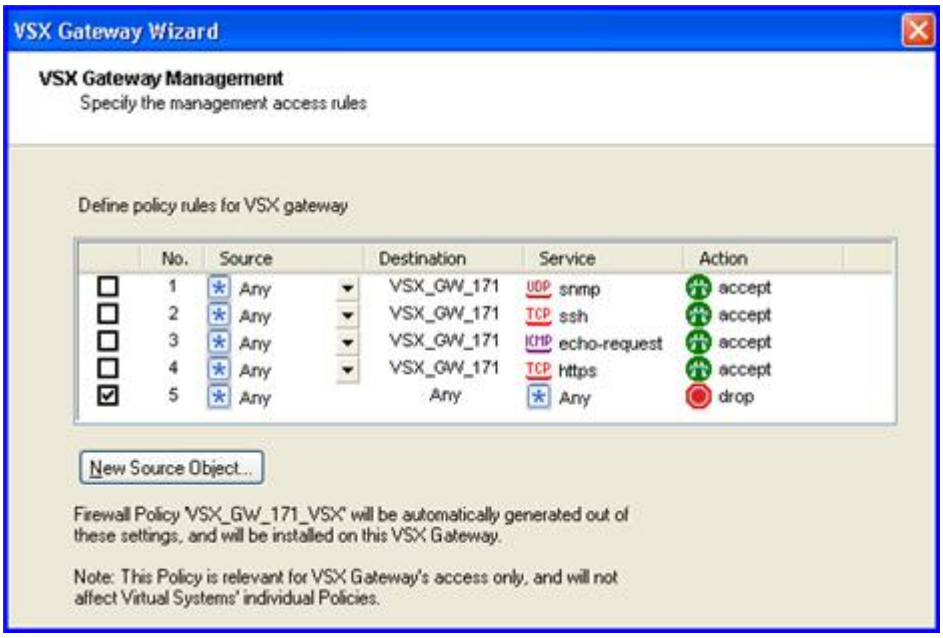
Important - This setting cannot be changed after you complete the VSX Gateway Wizard. If you define a non-DMI gateway, you cannot change it to a DMI gateway later.

4. Define the **IP address** and **Net Mask** for a Virtual Router.
   These options are not available for a Virtual Switch.
5. Optionally, define a **Default Gateway** for a Virtual Router (DMI only).

VSX Gateway Management

In the **VSX Gateway Management** window, define security policy rules that protect the VSX gateway. This policy is installed automatically on the new VSX gateway.

Note - This policy applies **only** to traffic destined for the VSX gateway. Traffic destined for Virtual Systems, other Virtual Devices, external networks, and internal networks is not affected by this policy.


VSX Gateway Wizard
VSX Gateway Management
Specify the management access rules
Define policy rules for VSX gateway
No.
Source
Destination
Service
Action
1 Any VSX_GW_171 UDP snmp accept
2 Any VSX_GW_171 TCP ssh accept
3 Any VSX_GW_171 ICMP echo-request accept
4 Any VSX_GW_171 TCP https accept
5 Any Any Any drop
New Source Object...
Firewall Policy 'VSX_GW_171_VSX' will be automatically generated out of these settings, and will be installed on this VSX Gateway.
Note: This Policy is relevant for VSX Gateway's access only, and will not affect Virtual Systems' individual Policies.

The security policy consists of predefined rules for these services:

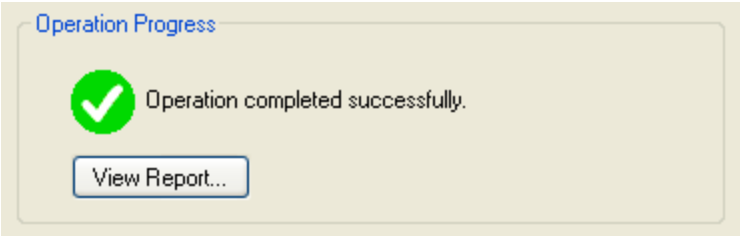
- **UDP -** snmp requests
- **TCP -** ssh traffic
- **ICMP -** echo-request (ping)
- **TCP -** https (secure http) traffic

Configuring the Gateway Security Policy

1. **Allow**: Select to pass traffic on the selected services. Clear this option to block traffic on this service. By default, all services are blocked.
   For example, to be able to ping the gateway from the management server, allow ICMP echo-request traffic.
2. **Source**: Click the arrow and select a **Source Object** from the list.
   The default value is ***Any**. Click **New Source Object** to define a new source.

Completing the VSX Wizard

Click **Next** to continue and then click **Finish** to complete the VSX Gateway wizard. This may take several minutes to complete. A message shows successful or unsuccessful completion of the process.



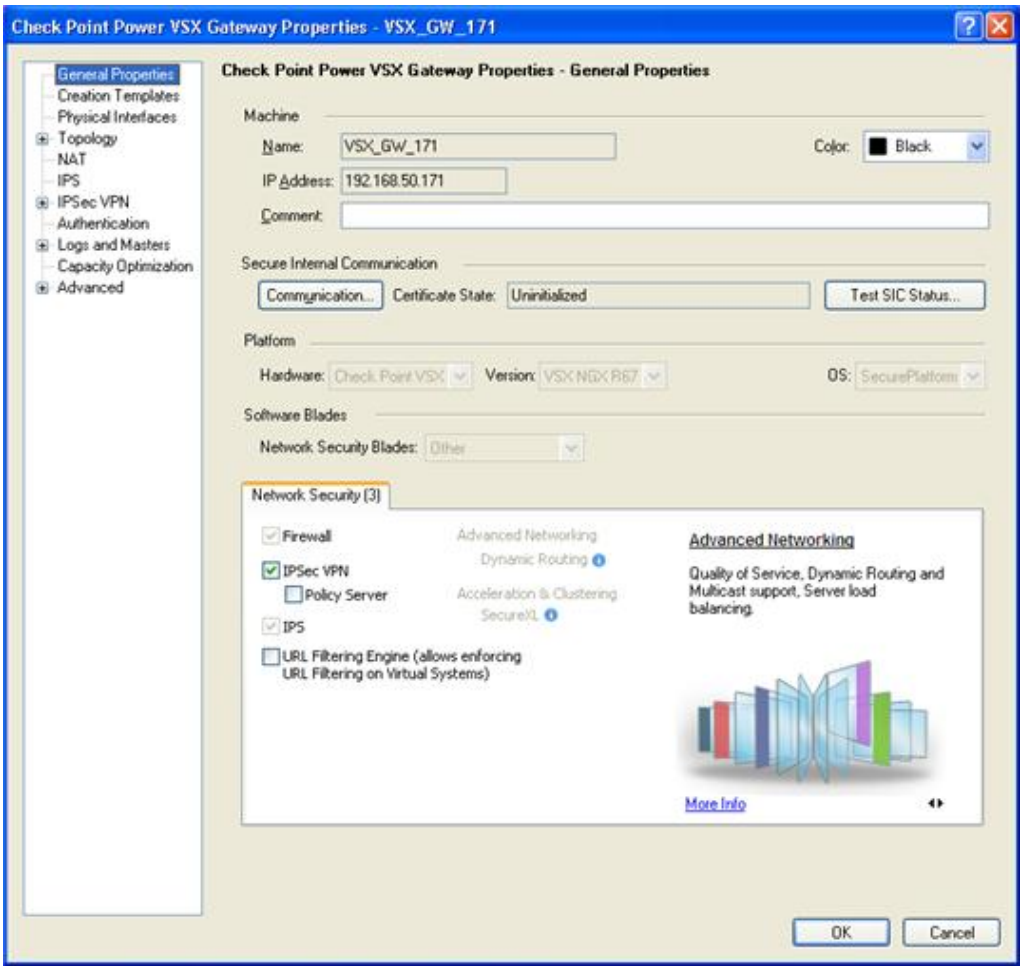
If the process ends unsuccessfully, click **View Report** to see the error messages. See the Troubleshooting chapter ("VSX Diagnostics and Troubleshooting" on page 179).

Modifying VSX Gateway Definitions

After you create a VSX gateway, you can modify the topology, other parameters, and advanced configurations in the **VSX Gateway Properties** window. To open this window, double-click on the VSX gateway object in the SmartDashboard Object Tree. The **VSX Gateway Properties** window opens, showing the **General Properties** page.

VSX Gateway - General Properties

In **General Properties**, check and re-establish SIC trust, and activate Check Point products for this VSX gateway.



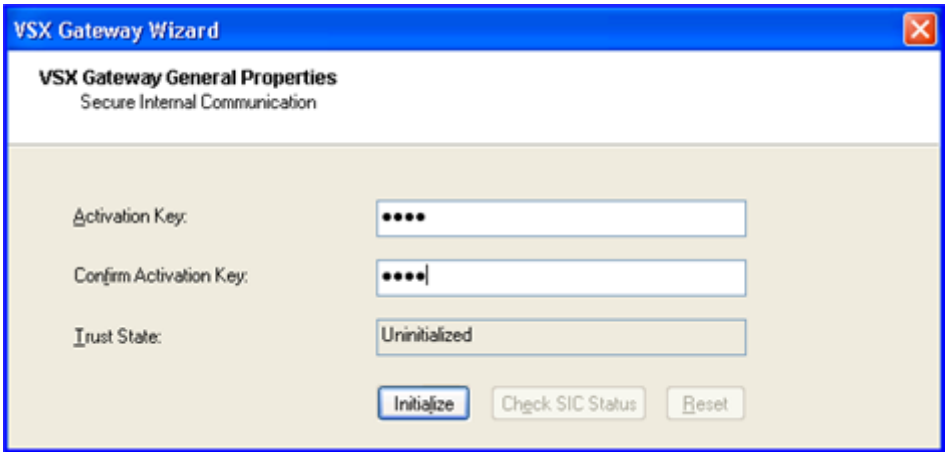
You can change these properties:

- **Comment** - Free text description for the Object List and elsewhere.

- **Color** - Color of the object icon as it appears in the Object Tree.
- **Secure Internal Communication** - Check and re-establish SIC trust.
- **Check Point Products** - Select Check Point products for this gateway.

Secure Internal Communication (SIC)

Test and reset SIC trust and also see the VSX gateway Relative Distinguished Name. To manage SIC, click **Communication**. The **Trusted Communication** window opens.



To initialize SIC trust, click **Initialize**.

If trust is not established successfully, click **Test SIC Status** to see the reason for the failure. The most common issues are an incorrect activation key and connectivity problems between the management server and the VSX gateway.

To reset SIC trust with the VSX gateway:

1. From the VSX gateway command line, use the **cpconfig** utility to re-initialize the SIC for the VSX gateway.
2. In the **Communication** window, click **Reset**.
3. Click **Yes** in the confirmation window.
4. Enter and confirm the SIC activation key in the appropriate fields.
5. Click **Initialize**.

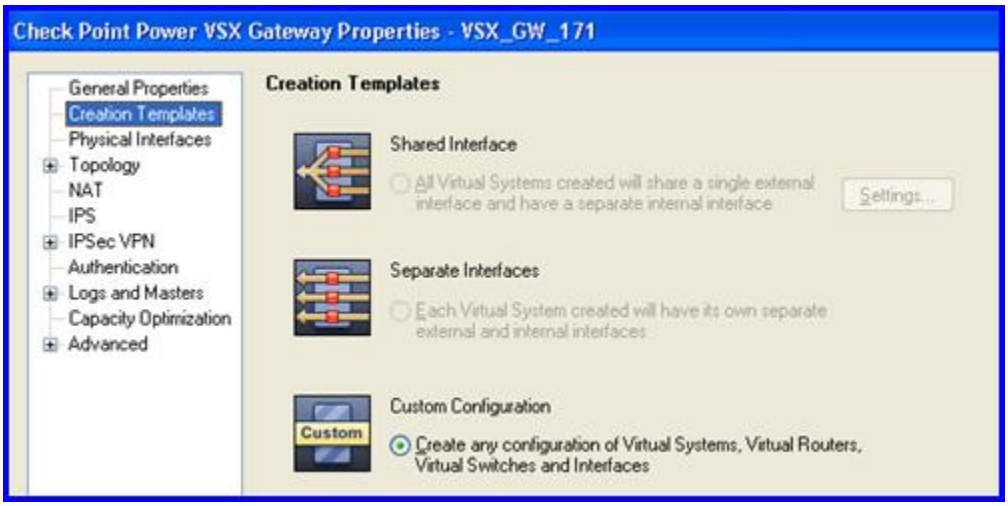
Check Point Products

Select the Check Point products to install on this Security Gateway from the list. The items you see are available for the product version and your license agreement.

firewall and the SVN Foundation are selected by default, because they are the essential product infrastructure. You cannot disable these items.

VSX Gateway - Creation Templates

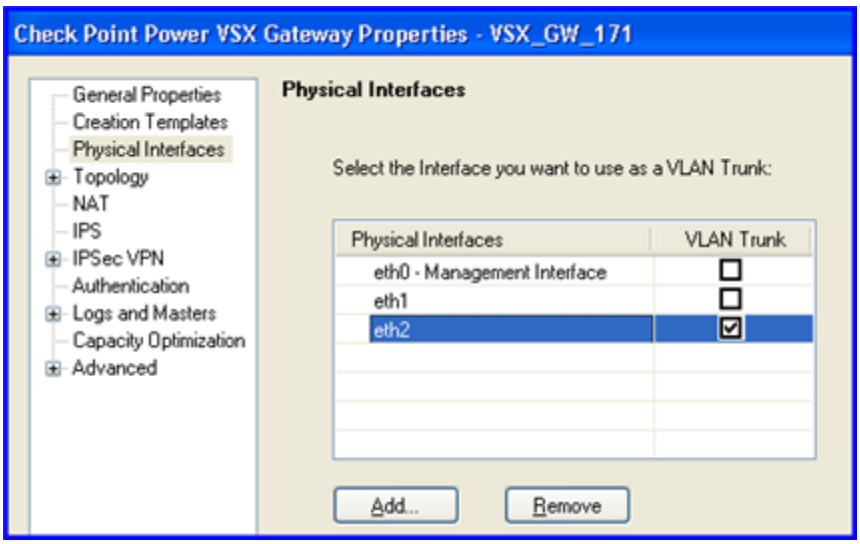
The **Creation Templates** page displays the creation template used to create the virtual systems for this Security Gateway. You can change from the current creation template to the **Custom Configuration** template and change the shared physical interface if the Shared Interface template is active.



- Select **Custom Configuration** to change from the Shared Interfaces or Separate Interfaces templates. This effectively overrides the default template. You cannot change back from the **Custom Configuration** template once you have completed the definition and saved it to the configuration to Security Gateway.
- To change the shared interface, click **Settings** and select an interface.

VSX Gateway - Physical Interfaces

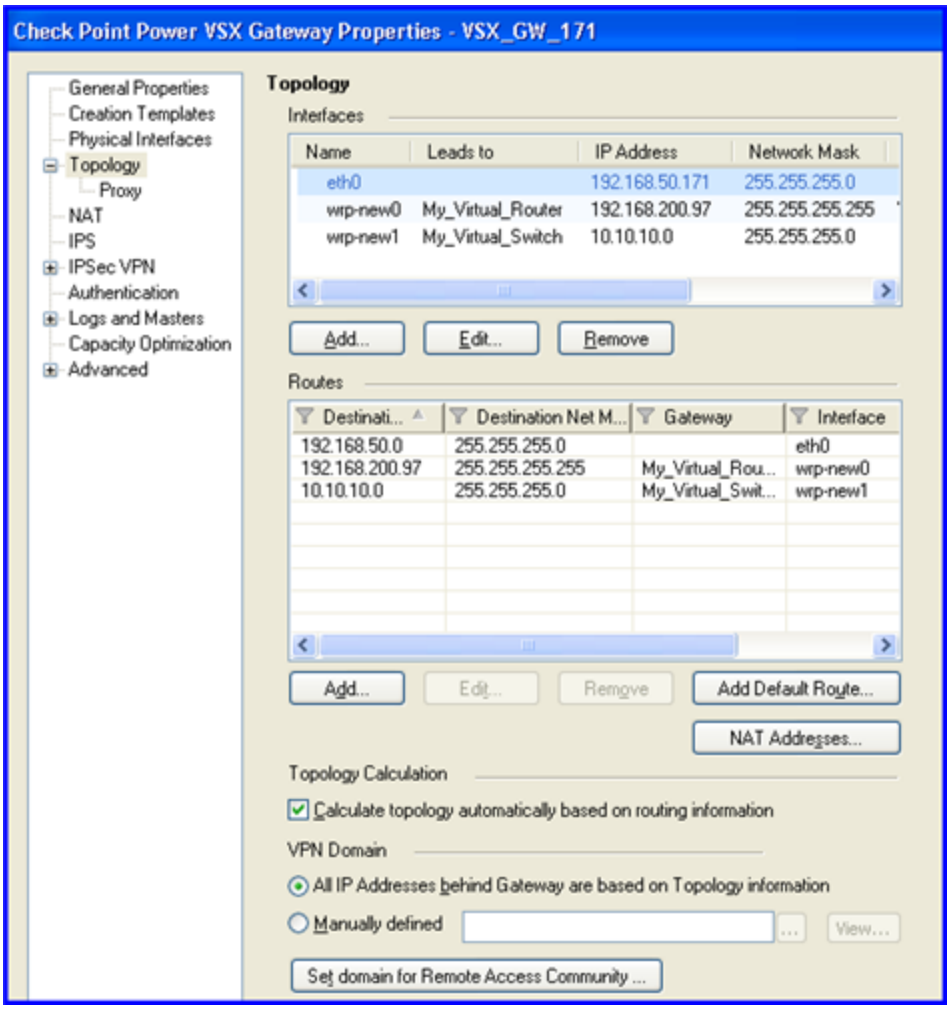
The **Physical Interfaces** page allows you to add or delete a physical interface on the VSX gateway, and to define interfaces to be used as VLAN trunks.



- To add a new physical interface, click **Add** and enter the interface name in the appropriate field.
- To define an interface as a VLAN trunk, select the desired interface and enable the check box. To disable a VLAN trunk, clear the check box.

VSX Gateway - Topology

The **Topology** page contains definitions for interfaces and routes between interfaces and virtual devices.



Interfaces

The Interfaces section defines interfaces and links to devices. You can add new interfaces as well as delete and modify existing interfaces.

To add an interface, click **Add**. The **Interface Properties** window opens. Select an interface from the list and define the appropriate properties ("Modifying an Interface Definition" on page 63).

Routes

The Routes section defines routes between network devices, network addresses, and virtual devices. Some routes are defined automatically based on the interface definitions. You can add new routes as well as delete and modify existing routes.

To add a default route to the routing table, click **Add Default Routes** and either enter the default route IP address or select the default Virtual Router. The Route Configuration window opens. Click **Help** for details regarding the various properties and options.

Calculating topology automatically based on routing information

Enable this option to allow VSX to automatically calculate the network topology based on interface and routing definitions (enabled by default). VSX creates automatic links, or connectivity cloud objects linked to existing internal or external networks.

- This option is not available in the Bridge Mode.
- When employing dynamic routing, it is recommended to disable this option.

Note - If you wish to enable anti-spoofing protection when there are no routes pointing to internal networks, disable the **Calculating topology** option and modify the appropriate interface definitions to enable anti-spoofing.

VSX Gateway - NAT

This page contains various NAT options that are not relevant for VSX gateways.

VSX Gateway - VPN

The **VPN** page contains a variety of configuration properties for VSX gateways in site-to-site VPN deployments. This window is only available if the Check Point VPN product is enabled on the **General Properties** page.

Please refer to the online help and the *R75 VPN Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11675) for further details regarding VPN concepts and configuration.

VSX Gateway - Remote Access

The Remote Access page contains properties that govern establishing VPN connections with Remote Access clients. This window is only available if the Check Point VPN product is enabled on the **General Properties** page.

Please refer to the online help and the *R75 VPN Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11675) for further details regarding VPN with Remote Access clients.

VSX Gateway - Authentication

The **Authentication** page allows you to enable several different authentication options for a VSX gateway. See Authentication ("Working with Authentication" on page 63) for further details.

VSX Gateway - Logs and Masters

The **Logs and Masters** page allows you define logging options for a VSX gateway. Refer to configuration procedures ("Tracking Activity with SmartView Monitor" on page 69) for further details.

VSX Gateway - Capacity Optimization

The Capacity Optimization page allows you to maximize VSX gateway and VPN throughput by limiting the number of concurrent connections to the VSX gateway, the number of concurrent IKE negotiations, and the number of concurrent VPN tunnels.

To raise or lower the maximum, use the arrows in the appropriate field to set the desired value.

VSX Gateway - Advanced Pages

There are several configuration options for SNMP, connection persistence and permissions to install policies. To learn more, see to the online help and the*R75.20 Firewall Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12267) and *R75.20 IPS Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12270).

Deleting a VSX Gateway

Deleting a VSX gateway object automatically deletes all Virtual Systems and other virtual devices associated with that gateway. To delete a VSX gateway, right click the VSX gateway object on the Object Tree and click **Delete**. Select **Yes** in the confirmation box.

VSX Gateway Recovery

In the event of a catastrophic VSX gateway failure, you can use the `vsx_util` command to restore the VSX gateway configuration as well as its virtual device configuration.

1. Reinstall the gateway and configure IP, net mask and default gateway.
2. Verify that all management interfaces have the same IP addresses as before.
3. From a command line interface on the management server, run `vsx_util reconfigure` to restore the previous configuration.

Working with Virtual Systems

This section presents procedures for creating and configuring Virtual Systems. The Virtual System definition process varies somewhat according to the template selected when creating the VSX gateway.

A typical Virtual System contains two interfaces:

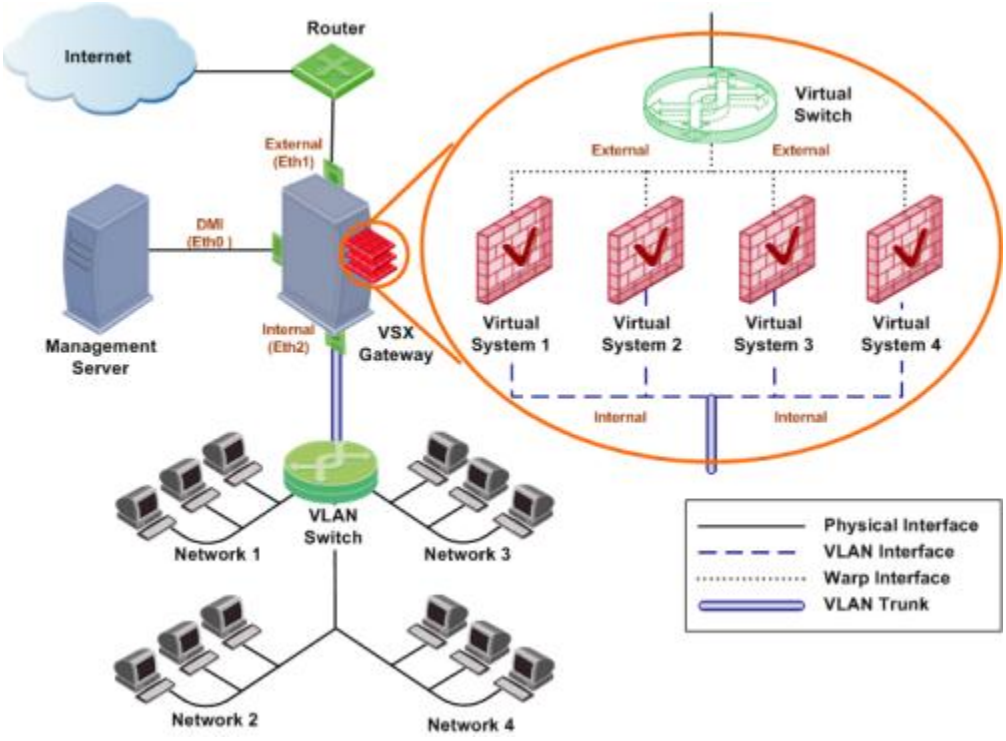
- **External interface** leading to external networks, a DMZ, or the Internet
- **Internal interface** leading to internal networks or servers, often by means of a VLAN trunk

VSX supports up to 64 interfaces per virtual device and a total of up to 4096 interfaces per gateway or cluster.

You can add as many interfaces to a Virtual System as required by your environment, subject to system resource limitations.

The following illustration illustrates an example of a typical VSX gateway deployment with four Virtual Systems, each containing two interfaces.

Figure 3-14 Typical VSX deployment



Creating a New Virtual System

You use the **Virtual Systems Wizard** to create a new Virtual System. You can modify the initial definition and configure advanced options after completing the wizard. The procedure consists of the following two steps, each represented by a wizard window:

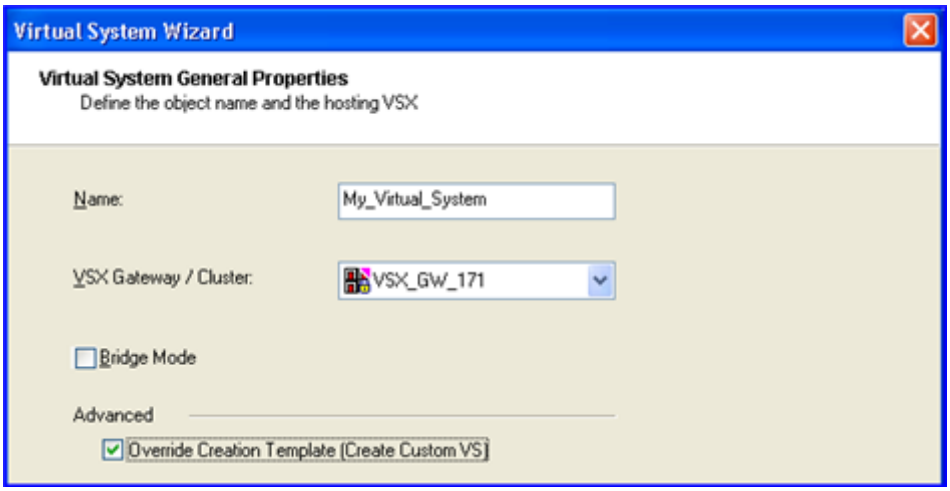
To start the VSX Gateway wizard**:**

1. Open SmartDashboard. If you are using Multi-Domain Security Management, open SmartDashboard from the Domain Management Server in which you are creating the Virtual System.

2. In the **Network Objects** tab, located in the **Objects Tree**, right-click **Check Point** and select **New Check Point > VPN-1 Power VSX > Virtual System**. The **VSX Gateway Wizard** opens, displaying the **General Properties** page.

Defining General Properties

The **General Properties** wizard page contains properties that define the Virtual System object and the hosting VSX gateway.



This window contains the following properties:

- **Name**: Unique, alphanumeric for the VSX gateway. The name cannot contain spaces or special characters except the underscore.
- **VSX Cluster/Gateway**: Select the VSX gateway hosting the Virtual System.
- **Bridge Mode**: Enable this option to create a Virtual System in the Bridge Mode.
- **Override Creation Template**: You can optionally override the creation template selected during the initial configuration of the VSX gateway. This allows you to add additional interfaces and to modify the topology and other properties.

Defining Network Configuration

The Virtual System Network Configuration page allows you to define internal and external interfaces as well as the IP address topology located behind the internal interface. The process for Virtual System defining network properties varies according to the several factors:

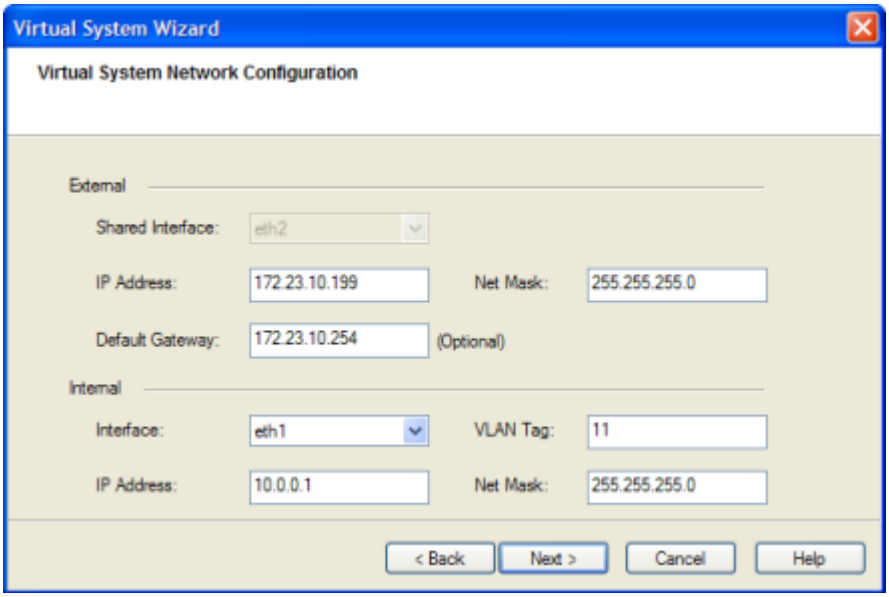
- The **VSX Gateway Creation** template used to define the gateway containing the virtual system.
- Whether or not you choose to override the default VSX Gateway Creation template. This has the effect of using the **Custom Configuration** template.
- Whether or not you create the Virtual System in the Bridge Mode.

Note - The Bridge mode is not available for a Virtual System created with the Shared Interface template.

Use one of the following network configuration scenarios, according to your environment:

Shared Interface or Separate Interfaces

The **Virtual System Network Configuration** page for the Shared Interface and Separate Interfaces templates appears as shown.

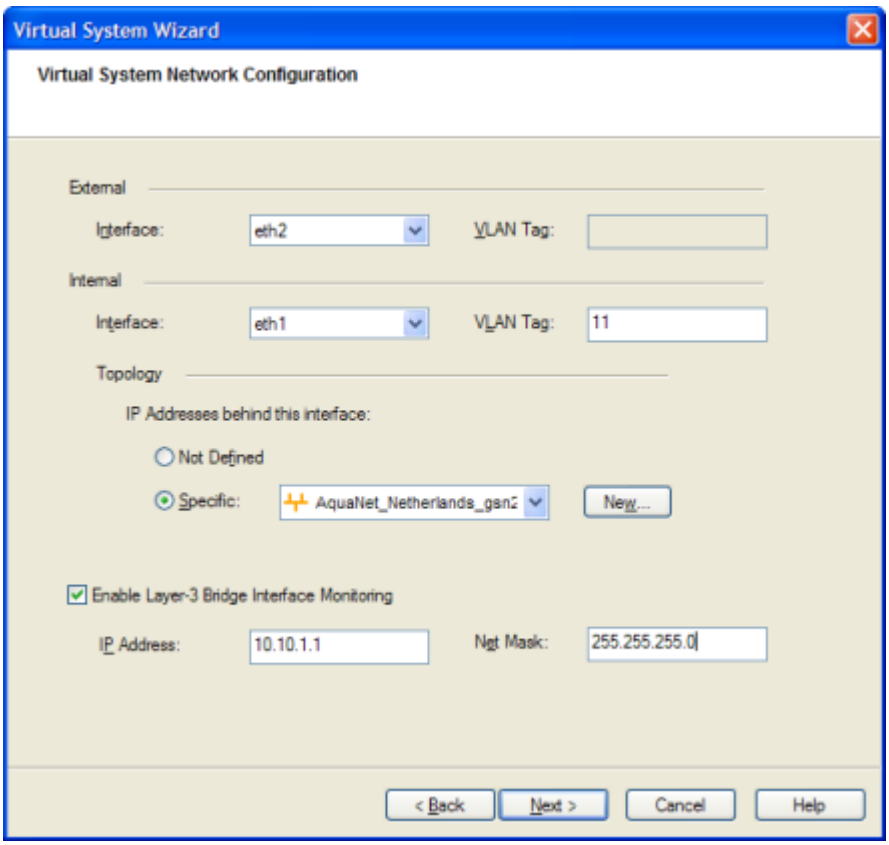


To configure the external and internal interfaces:

1. Select the desired interfaces from the appropriate list.
2. If the selected Interface is a VLAN interface, enter the VLAN tag in the appropriate field. This field is not available for non-VLAN interfaces.
3. Enter the IP address and net mask in the appropriate fields. Optionally, enter a default gateway for the external interface.
4. Complete the definition process ("Completing the Definition" on page 47).

Separate Interfaces in Bridge Mode

The **Virtual System Network Configuration** page for the Separate Interfaces template in the Bridge Mode appears as shown.



To configure the external and internal interfaces:

1. Select the desired interfaces for the internal and external networks from the appropriate list.

   If the selected Interface is a VLAN interface, enter the same VLAN tag in both the external and internal **VLAN Tag** fields. This field is not available for non-VLAN interfaces.
2. Define the topology for the internal interface as follows:
   - Select **Not Defined** if you do not wish to define an IP address.
   - Select **Specific** and then select a IP address definition from the list. IP address definitions can be based on object groups or predefined networks that define the topology.
3. If you wish to create a new IP address definition perform the following steps:

   a) Select **Specific** and click **New**.

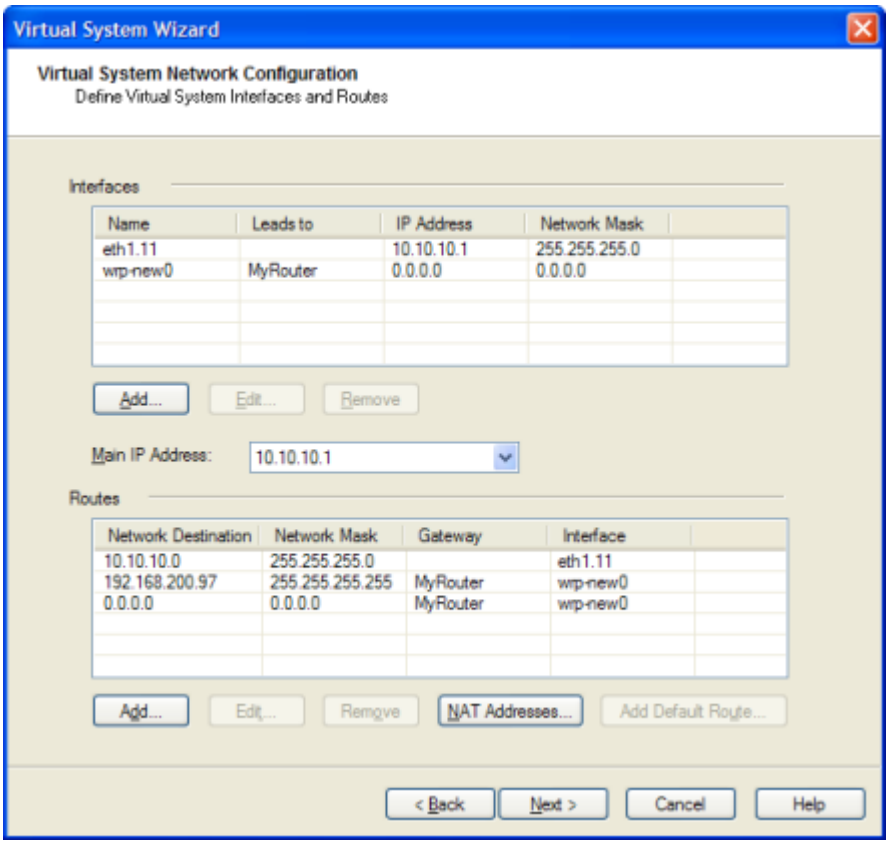
   b) Select **Group** to define an object group or **Network** to define network properties. The appropriate window appears. Refer to the online help for details regarding either of these options.
4. Enable the **Layer-3 bridge interface monitoring** option if you wish to enable layer 3 network fault detection for this Virtual System.

   a) Enter an IP address and subnet mask in the designated fields for this Virtual System, which continuously monitors the specified network for faults or connectivity issues. The IP address/subnet should define the network on which the Virtual System resides.

   Note - When creating a Virtual System in the bridge mode on an IPSO cluster, you must enable Layer-3 bridge interface monitoring. The IP address to be monitored should reside on a different subnet than the one that handles bridge traffic.
5. Complete the definition process ("Completing the Definition" on page 47).

Custom Configuration or Override - Non-Bridge Mode

If you used the Custom Configuration template when creating the VSX gateway, or if you selected the **Override Creation Template** option, it is necessary to manually define the network interfaces and connections. The **Virtual System Network Configuration** page for Custom Configuration appears as shown:



To configure the external and internal interfaces:

1. In the interface table, define interfaces. You can add new interfaces as well as delete and modify existing interfaces.

   To add an interface, click **Add**. The **Interface Properties** window opens. Select an interface from the list and define the appropriate properties. Click **Help** for details regarding the various properties and options.
2. Select the **Main IP Address** from the list. This IP address, typically that assigned to the external interface, specifies the "real" Virtual System address used when working with NAT or VPN connections.

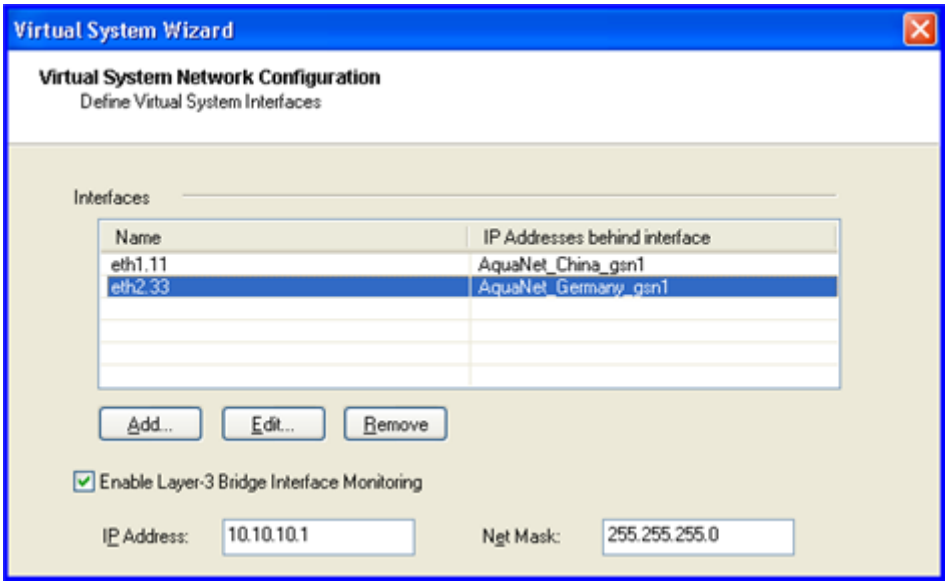
   To make your external IP address routable, select the external interface IP address as the main IP address.
3. Define network routing as appropriate for your deployment. Some routes are automatically defined automatically based on the interface definitions.

   For example, you would generally define a default gateway route leading to an external Virtual Router or to the Virtual System external interface.

   To add a default route to the **Routes** table, click **Add Default Routes** and either enter the default route IP address or select the default Virtual Router. The Route Configuration window opens. Click **Help** for details regarding the various properties and options.
4. Complete the definition process ("Completing the Definition" on page 47).

Custom Configuration or Override in the Bridge Mode

If you used the **Custom Configuration** template when creating the VSX gateway, or if you selected the **Override Creation Template** option, and are creating a Virtual System in the Bridge Mode, you will need to manually define the network interfaces. The **Virtual System Network Configuration** page appears as shown.



- **Interfaces**: To configure the external and internal interfaces, define interfaces and links to devices in the **Interfaces** table. You can add new interfaces as well as delete and modify existing interfaces.

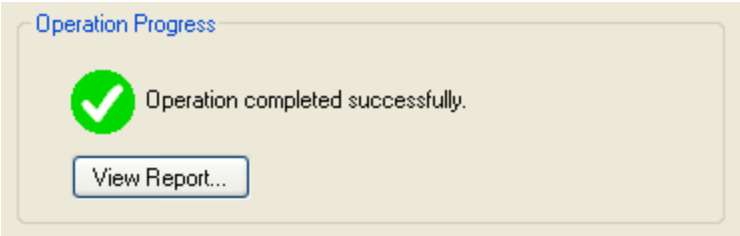
  To add an interface, click **Add**. The **Interface Properties** window opens. Select an interface from the list and define is properties. Click **Help** for details regarding the various properties and options.

- **Layer-3 Bridge Interface Monitoring**: This option only appears for Virtual Systems hosted by clusters on Nokia platforms. Enable this option to monitor interface traffic at the IP address and net mask specified in the designated fields.

  When creating a Virtual System in the bridge mode on a Nokia platform, you **must** enable layer-3 bridge interface monitoring. The IP address to be monitored should reside on a different subnet than the subnet that handles bridge traffic.

Completing the Definition

Click **Next** and then **Finish** to create the Virtual System. Please note that this may take several minutes to complete. A message appears indicating successful or unsuccessful completion of the process.



If the process ends unsuccessfully, click **View Report** to view the error messages. Refer to the troubleshooting chapter ("VSX Diagnostics and Troubleshooting" on page 179) for further assistance.

Once you create a VSX gateway using the VSX Wizard, you can modify the topology and all other parameters using the **VSX Gateway Properties** window. This window also allows you to access many advanced features and options that are not available via the wizard.

Modifying a Virtual System Definition

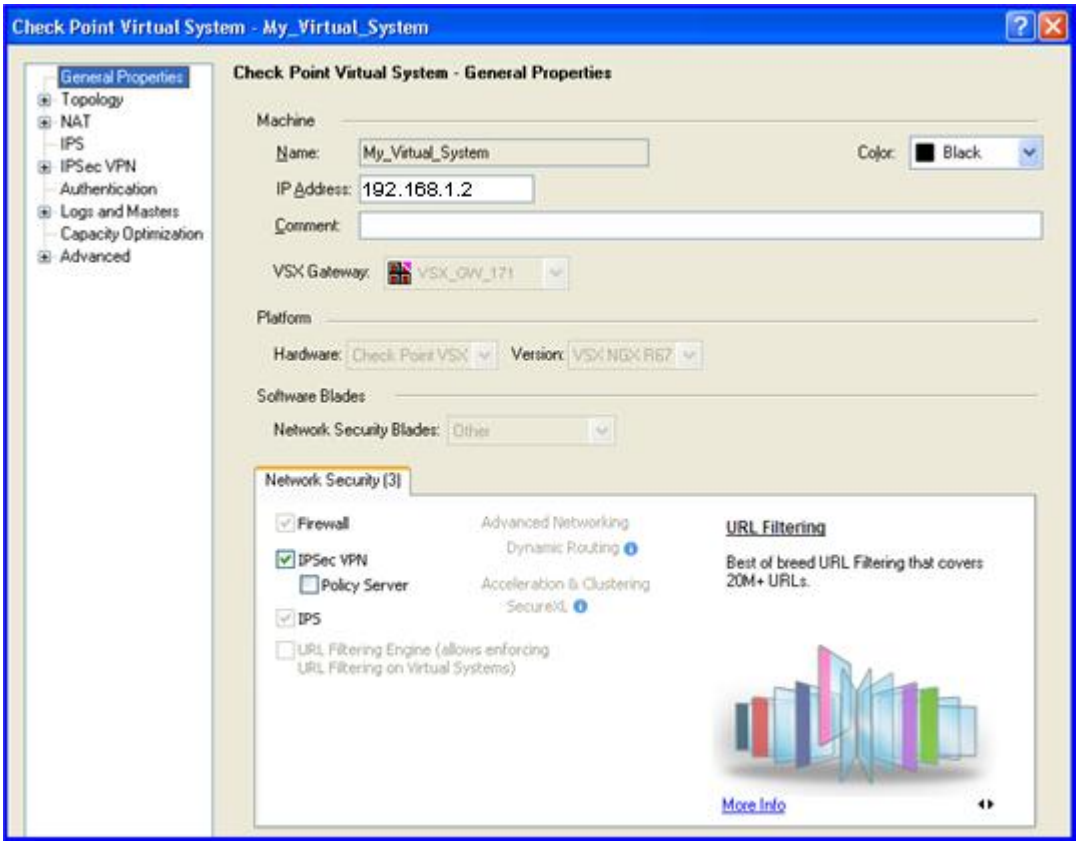
Once you create a Virtual System using the wizard, you can modify the topology and other properties using the **Check Point Virtual System** window. This window also allows you to configure many advanced features and options that are not available in the wizard.

To work with a Virtual System definition, double-click the Virtual System object in the Object tree. The **Check Point Virtual System** window opens, displaying the General Properties page.

The following sections describe the various pages and properties that constitute a Virtual System definition.

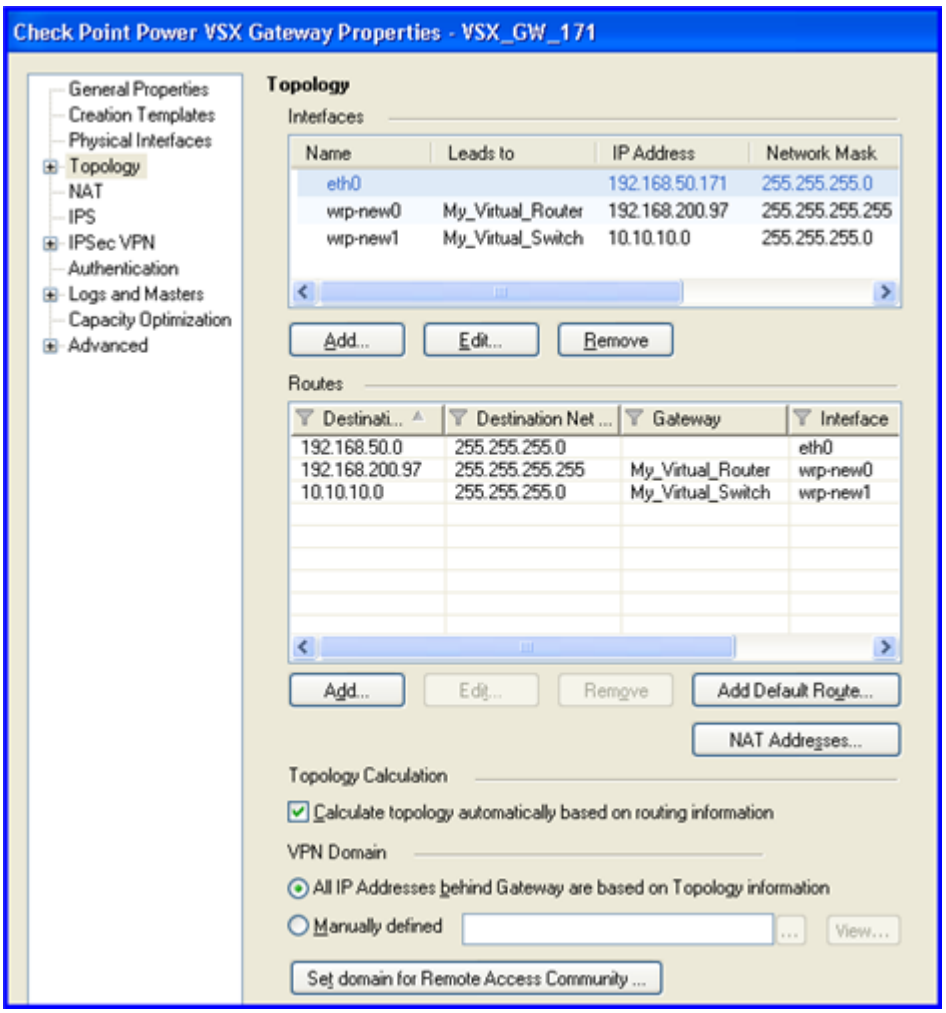
Virtual System - General Properties

The **General Properties** page allows you specify the main IP address and to enable various Check Point products for a Virtual System.



Virtual System - Topology

The **Topology** page contains definitions for Virtual System interfaces, routes and Warp links. Based on these interface settings, VSX automatically creates routes to Virtual Devices and the VSX gateway.



Note - If you modify the topology for a specific Virtual System in a cluster environment, the cluster topology is not updated until you install a policy on that Virtual System.

- **Interfaces**: The Interfaces table defines interfaces and links to devices. You can add new interfaces as well as delete and modify existing interfaces.
- To add an interface, click **Add**. The **Interface Properties** window opens. Select the interface from the list and define the appropriate properties. The Modifying an Interface Definition section ("Modifying an Interface Definition" on page 63) and the online help provides explanations of the various properties and options.
- **Routes**: To add a default route to the Routes table, click **Add Default Routes** and either enter an IP address or select a Virtual Router. The **Route Configuration** window opens. Click **Help** for details regarding the various properties and options.
- **Calculate topology automatically based on routing information:** Enable this option to allow VSX to automatically calculate the network topology based on interface and routing definitions (enabled by default). VSX creates automatic links, or connectivity cloud objects linked to existing internal or external networks.
  - When this option is enabled, you cannot configure the topology using **Topology** tab in the **Interface Properties** window. These options are unavailable on the tab.
  - This option is not available in the Bridge Mode.
  - When employing dynamic routing, it is recommended to disable this option.

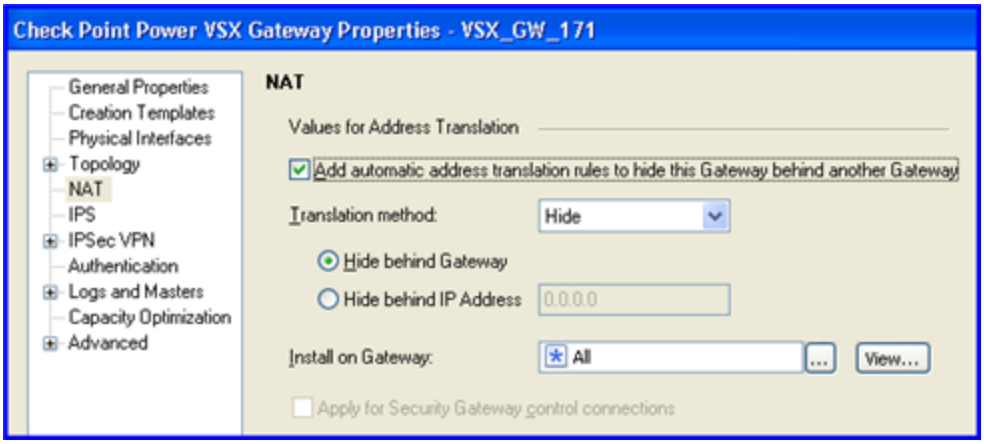
- **VPN Domain**: The VPN Domain defines the set of hosts located behind a given Virtual System that communicate via a VPN tunnel with peer Virtual Systems. These options are only available if you selected **VPN** in the **Check Point Products** section on the **General Properties** page.

  When including a virtual device as part of a VPN connection, you must specify a VPN Domain. The domain definition specifies Virtual System interfaces that are included in the VPN. You can define a VPN Domain in one of two ways by enabling the appropriate option:

- **All IP Addresses behind gateway based on topology information**: Includes all hosts not located behind an external gateway cluster interface.
- **Manually Defined**: Includes all hosts in the selected network or group.

Virtual System - NAT

The **NAT** page allows you to configure NAT rules for packets originating from a Virtual System.



To enable and configure NAT for a Virtual System:

1. Enable the **Add Automatic Address Translation** option.
2. Select a translation method from the list.
   - **Hide NAT**: Hide NAT only allows connections originating from the internal network. Internal hosts can access internal destinations, the Internet and other external networks. External sources cannot initiate a connection to internal network addresses.
   - **Static NAT**: Static NAT translates each private address to a corresponding public address.
3. If you select **Hide NAT**, select one of the following options:
   - **Hide behind Gateway** hides the real IP address behind the virtual system external interface IP address,

     or

   - **Hide behind IP Address** hides the real address behind a virtual IP address, which is a routable, public IP address that does not belongs to any real machine.
4. If you selected **Static NAT**, enter the static IP address in the appropriate field.
5. Select the desired VSX gateway from the **Install on Gateway** list.

Virtual System - IPS

Virtual Systems use the default protection profile. There are no configurable options here.

Virtual System - VPN

The **VPN** page contains a variety of configuration properties for Virtual Systems in site-to-site VPN deployments. This window is only available if the Check Point VPN product is enabled on the **General Properties** page.

Please refer to the online help and the *R75 VPN Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11675) for further details regarding VPN concepts and configuration.

Virtual System - Remote Access

The Remote Access page contains properties that govern establishing VPN connections with Remote Access clients. This window is only available if the Check Point VPN product is enabled on the **General Properties** page.

Please refer to the online help and the *R75 VPN Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11675) for further details regarding VPN with Remote Access clients.

Virtual System - Authentication

The **Authentication** page allows you to enable several different authentication options for a VSX gateway. The Authentication section ("Working with Authentication" on page 63) provides additional details.

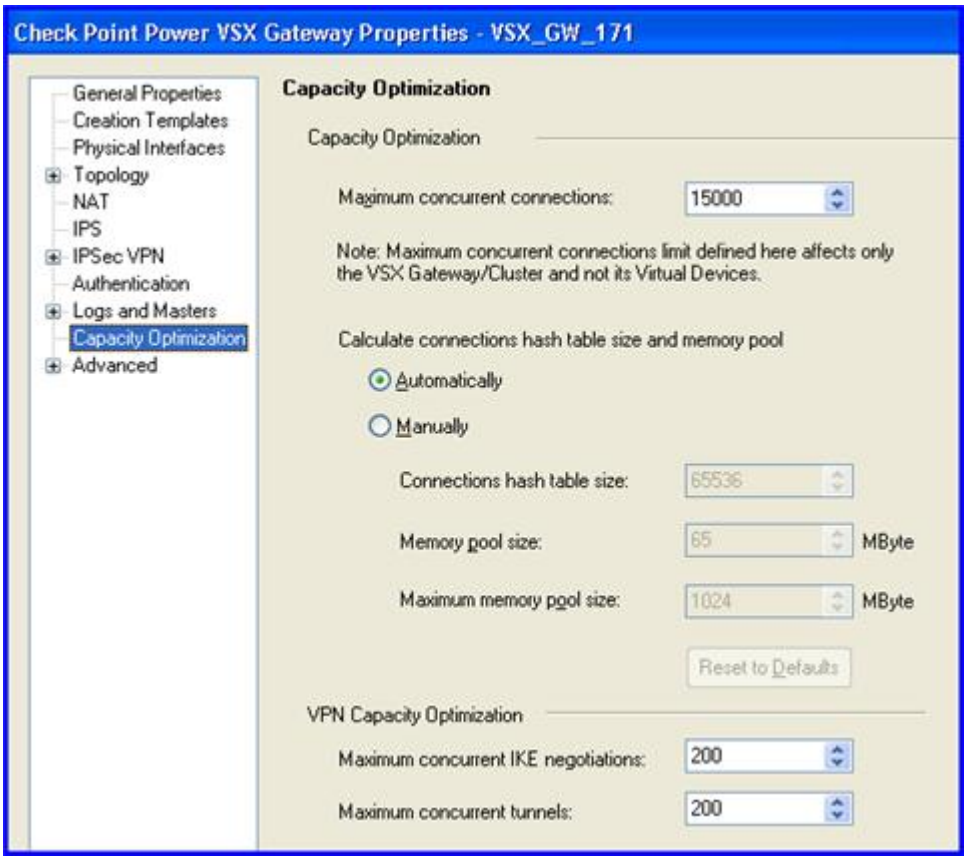
Virtual System - Logs and Masters

The **Logs and Masters** page allows you to define logging options for a VSX gateway. The SmartView Monitor ("Tracking Activity with SmartView Monitor" on page 69) section provides additional details for working with logs.

Virtual System - Capacity Optimization

The **Capacity Optimization** page allows you to maximize Virtual System and VPN throughput by limiting the following connection parameters:

- **Concurrent connections** (Default = 15,000)
- **Number of concurrent IKE negotiations** (Default = 200)
- N**umber of concurrent VPN tunnels** (Default = 200)



To raise or lower the maximum, use the arrows in the appropriate field to set the desired value. If you change the **Maximum concurrent connections** option, you must install a policy to the Virtual System.

Virtual System - Advanced

These pages contain a variety of configuration options for SNMP, connection persistence and permissions to install policies. For further information regarding these options, please refer to the online help and the *R75 IPS Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11663).

Deleting a Virtual System

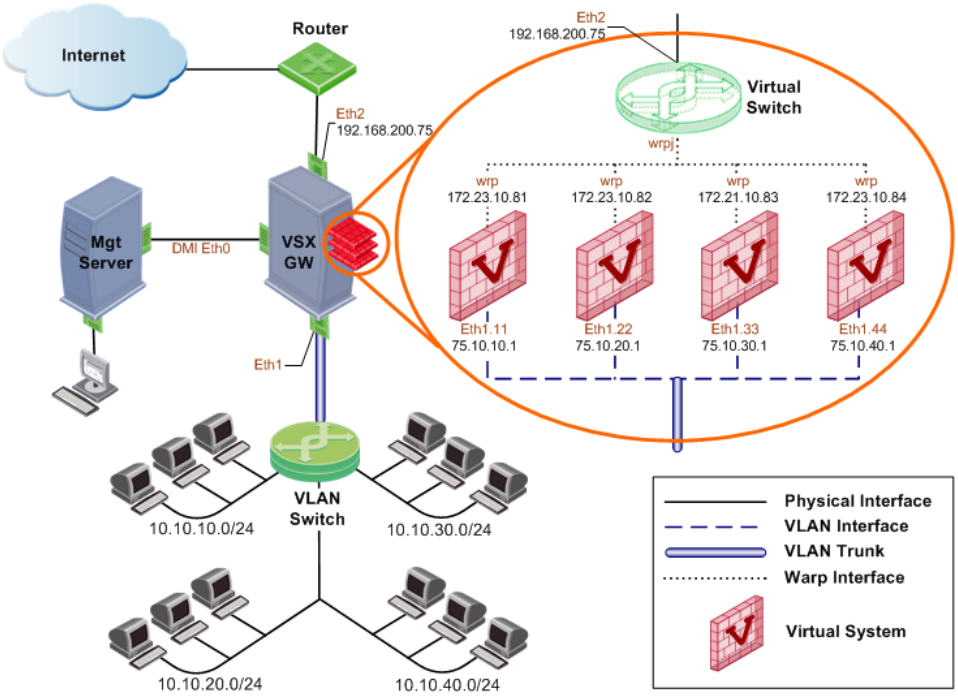
To delete a Virtual System, right-click the appropriate Virtual System object on the Object Tree and select **Delete**. Click **Yes** in the confirmation box.

Working with Virtual Switches

Virtual Switches provide level-2 connectivity between Virtual Systems and internal or external networks. This section describes how to define and configure a Virtual Switch. As with physical switches, each Virtual Switch maintains a forwarding table containing entries that describe known networks and directions for reaching them.

You can define Virtual Switches for external and internal communications.

Figure 3-15 Typical Virtual Switch deployment



The above figure shows a typical deployment using a Virtual Switch for external connections and a VLAN trunk leading to the internal, protected network.

Adding Virtual Switches

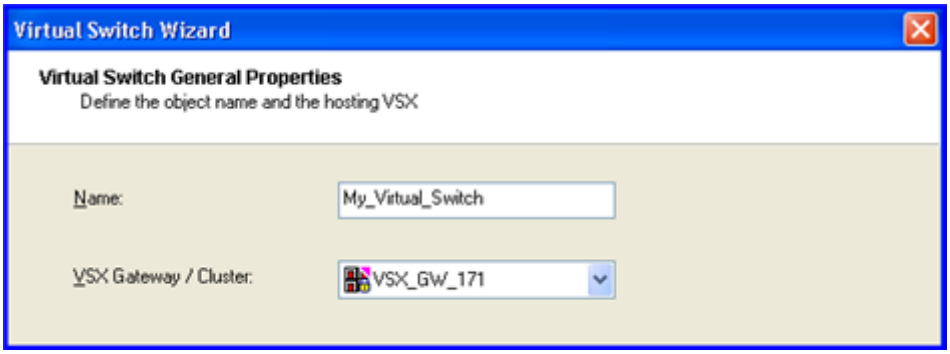
You use the **Virtual Switch Wizard** to create a new Virtual Switch. You can modify the initial definition and configure advanced options after completing the wizard. The definition procedure consists of two steps, each represented by a separate wizard window:

To create a new Virtual Switch using the wizard, perform the following steps:

1. Open SmartDashboard. If you are using Multi-Domain Security Management, open SmartDashboard from the Domain Management Server in which you are creating the Virtual Switch.
2. In the **Network Objects** tab, located in the **Objects Tree**, right-click **Check Point** and select **New Check Point > VPN-1 Power VSX > Virtual Switch**. The **Virtual Router Wizard** opens, displaying the **General Properties** page. The Virtual Switch Wizard opens.

Defining the General Properties

The **General Properties** page contains properties that identify the Virtual Switch and the VSX gateway or cluster to which it connects.

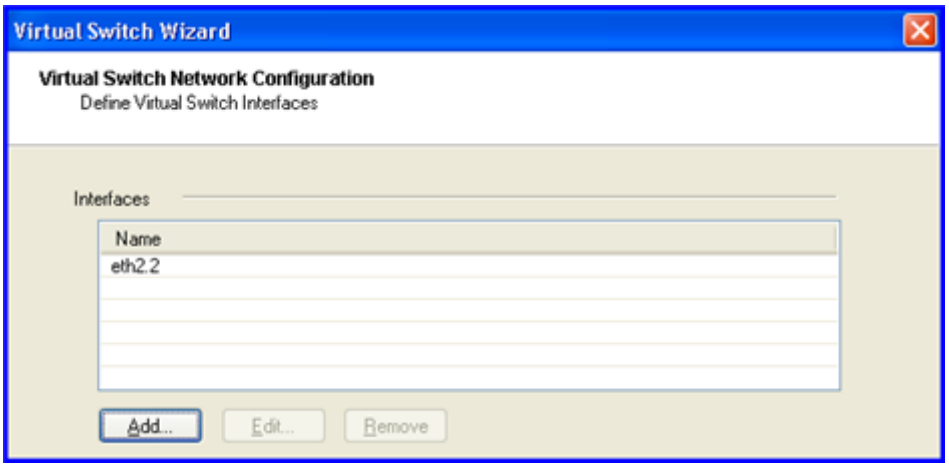


This window contains the following properties:

- **Name**: Unique name containing only alphanumeric characters, the hyphen and underscore characters.
- **VSX Gateway/Cluster**: Select a VSX gateway or cluster from the list.

Defining the Network Configuration

The **Network Configuration** page defines the Virtual Switch interface.



A Virtual Switch has one interface. Click **Add** and select the interface from the list. If applicable, enter a VLAN tag in the appropriate field.

Note - You can define a Virtual Switch with no interfaces for the purpose of communication between Virtual Systems.

Modifying Virtual Switches

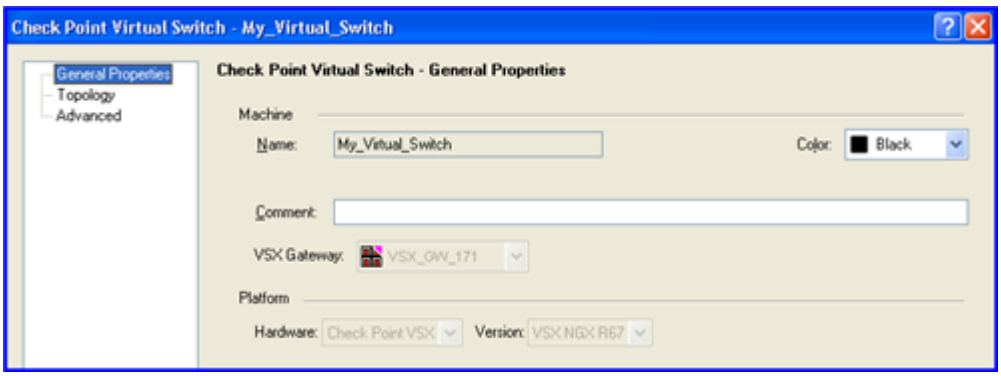
Once you create a Virtual Switch using the wizard, you can modify the topology and other properties using the **Check Point Virtual Switch** window. This window also allows you to configure advanced features and options that are not available in the wizard.

To work with a Virtual Switch definition, double-click the Virtual Switch object in the Object tree. The **Check Point Virtual Switch** window opens, displaying the **General Properties** page.

The following sections describe the various pages and properties that constitute a Virtual Switch definition.

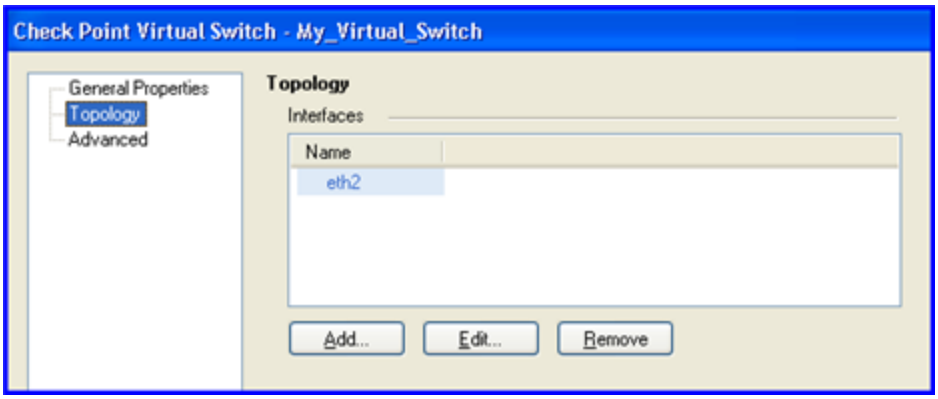
Virtual Switch - General Properties

The **General Properties** page allows you to add comments and change the icon color as displayed in SmartDashboard.



Virtual Switch - Topology

The **Topology** page defines Virtual Switch interfaces. You can only modify the single defined interface. Warp interfaces cannot be modified from this window.



To add an interface, click **Add**. The **Interface Properties** window opens. Select an interface from the list and define the IP address, net mask and other properties as required. Refer to the Modifying an Interface Definition section ("Modifying an Interface Definition" on page 63) or the online help for details regarding the various properties and options.

Deleting a Virtual Switch

You must remove all Virtual System connections before attempting to delete a Virtual Switch.

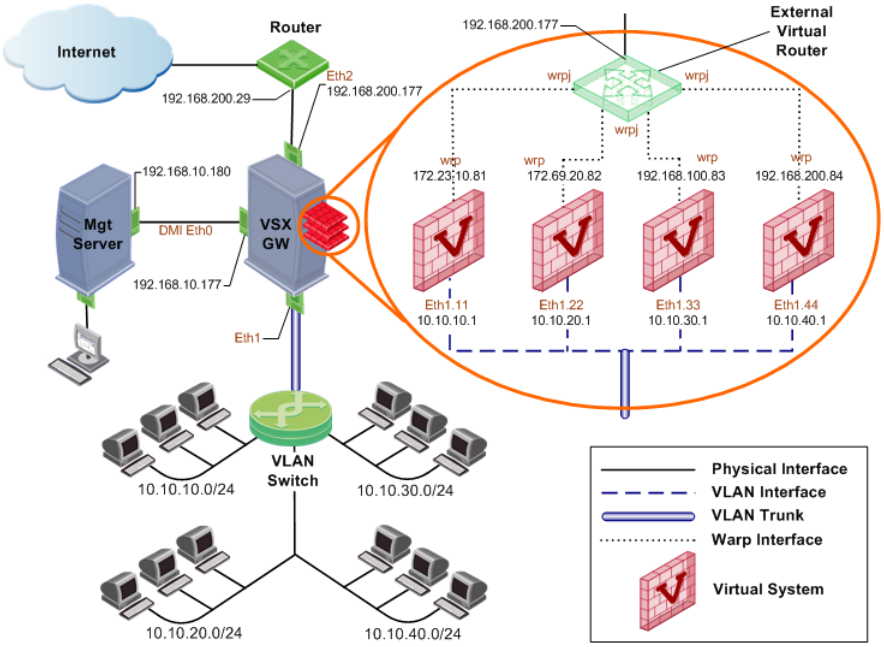
To delete a Virtual Switch, right-click the appropriate Virtual Switch object in the Object Tree and select **Delete**. Click **Yes** in the confirmation box.

Working with Virtual Routers

This section describes how to define and configure a Virtual Router. As with physical routers, each Virtual Router maintains a routing table containing entries that describe known networks and directions on how to reach them.

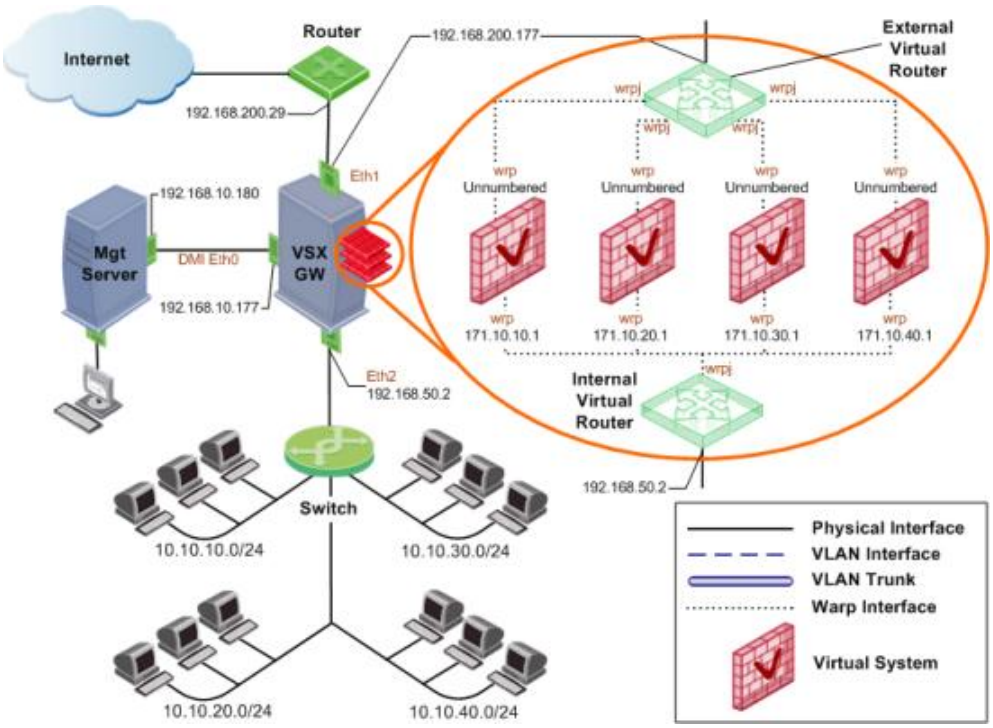
You can define Virtual Routers for both external and internal communications. A Virtual Router that connects to external networks, including a DMZ and the Internet, are referred to as an external Virtual Router. A Virtual Router that connects to internal, protected networks is known as an internal Virtual Router.

Figure 3-16 Deployment with an external virtual router



An external Virtual Router functions as the external gateway for Virtual Systems, allowing them to share a single secure physical interface leading to external networks and the Internet.

Figure 3-17 Source-based routing with Virtual Routers



In this scenario, VSX creates Warp interfaces between the Virtual Systems and both Virtual Routers. Note that the external Virtual System interfaces are defined as unnumbered interfaces.

An internal Virtual Router typically connects with one interface leading to internal networks through a switch with additional Warp Links leading to other Virtual Systems located in the VSX gateway.

The following sections describe how to create and configure Virtual Routers. The example screen shots refer to the topology illustrated above. After creating a new Virtual Router, you then add new interfaces to the Virtual Systems that connect to the newly created Virtual Router.

Creating a New Virtual Router

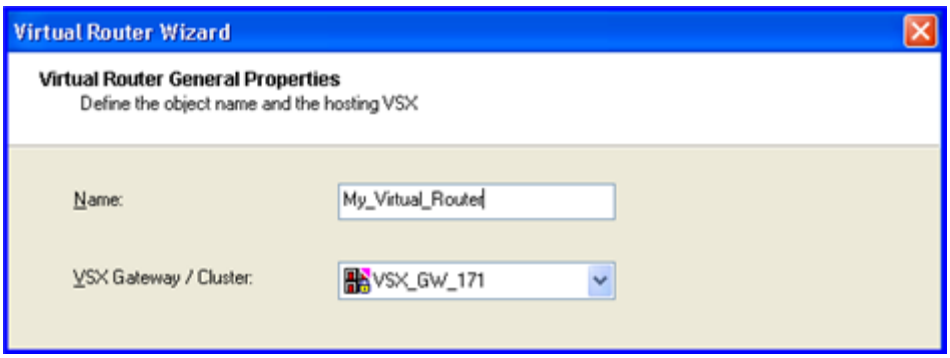
You use the **Virtual Router Wizard** to create a new Virtual Router. You can modify the initial definition and configure advanced options after completing the wizard. The definition procedure consists of two steps, each represented by a wizard window:

To create a new Virtual Router using the wizard:

1. Open SmartDashboard. If you are using Multi-Domain Security Management, open SmartDashboard to the Domain Management Server in which you are creating the Virtual Router.
2. In the **Network Objects** tab, located in the **Objects Tree**, right-click **Check Point** and select **New Check Point > VPN-1 Power VSX > Virtual Router**. The **Virtual Router Wizard** opens, displaying the **General Properties** page.

Defining General properties

The **General Properties** page contains properties that identify the router and the VSX gateway to which it connects.

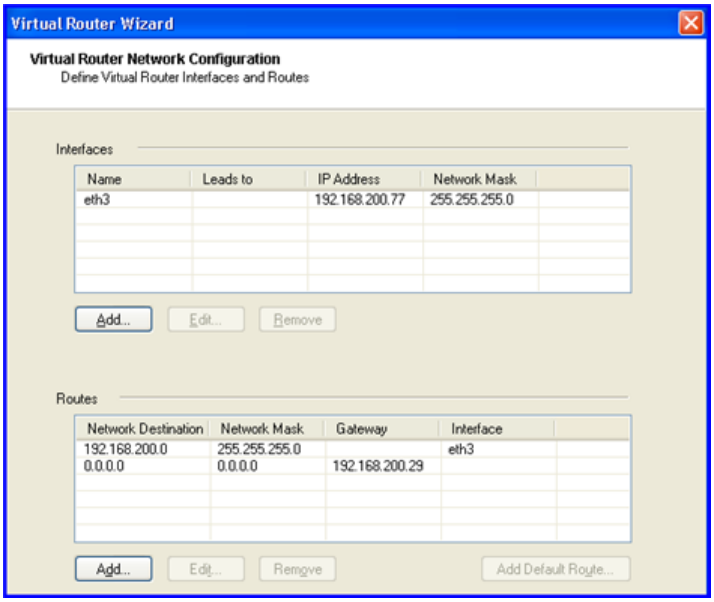


This window contains the following properties:

- **Name**: Unique name containing only alphanumeric characters and the hyphen and underscore characters
- **VSX Gateway/Cluster**: Select a VSX gateway or cluster from the list

Defining the Network Topology

The **Virtual Router Network Configuration** page defines the network topology for the Virtual Router. For an external interface, you define one or more shared external interfaces and a default gateway.



The topology definition consists of the following properties:

- **Interfaces**: Add new interfaces as well as modify or delete interfaces you have already defined.

To add an interface, click **Add**. The **Interface Properties** window opens. Select an interface from the list and define the IP address, net mask and other properties. Refer to Modifying an Interface Definition or the online help for details regarding the various properties and options.

- **Routes**: Add network routes between this Virtual router, Virtual Systems, external network devices and network addresses. Some Warp Link routes are defined automatically based on the interface definitions and cannot be modified or deleted. You can manually add new routes as well as delete and modify non-Warp Link routes.
- **Add Default Route**: Define the default gateway as an IP address or Virtual System.

Modifying a Virtual Router Definition

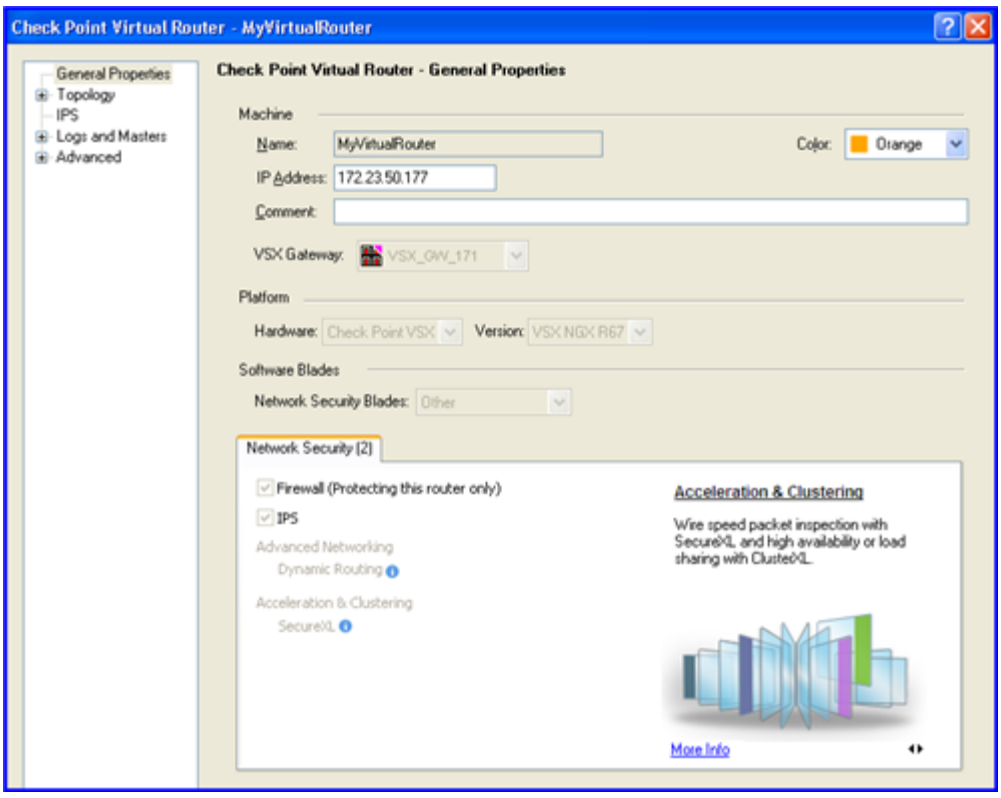
Once you create a Virtual Router using the wizard, you can modify the topology and other properties using the **Check Point Virtual Router** window. This window also allows you to configure many advanced features and options that are not available in the wizard.

To work with a Virtual Router definition, double-click the Virtual Router object in the Object tree. The **Check Point Virtual Router** window opens, displaying the General Properties page.

The following sections describe the various pages and properties that constitute a Virtual Router definition

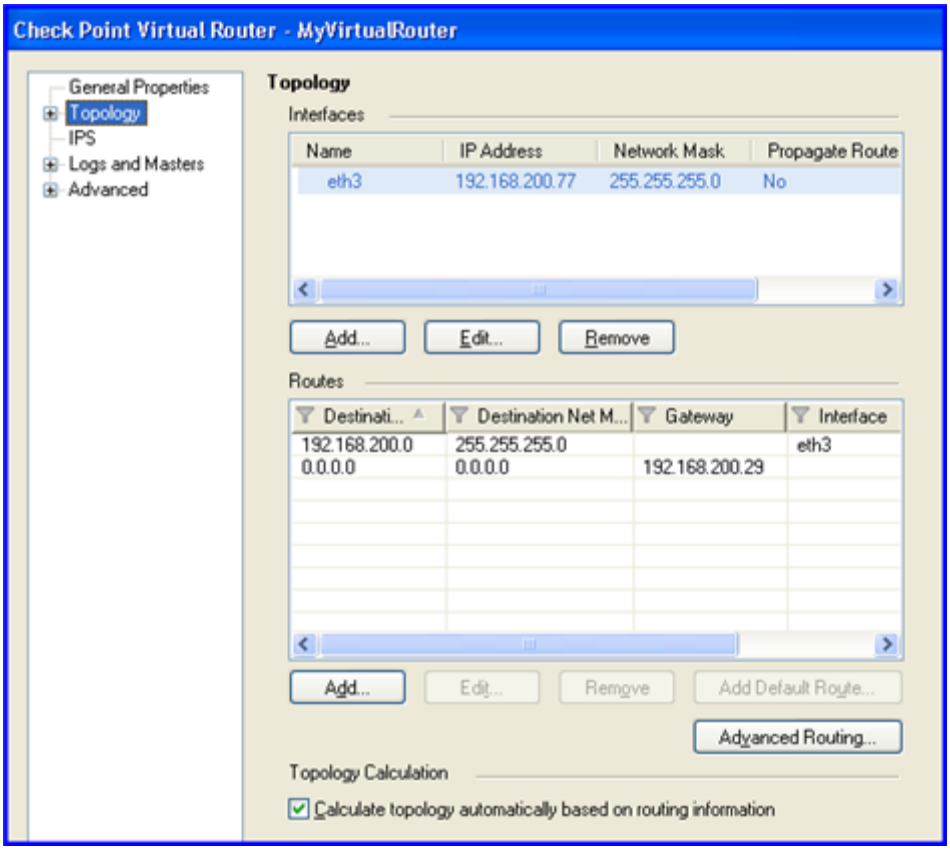
Virtual Router - General Properties

The **General Properties** page enables you change the Virtual router IP address as well as to add comments and change the icon color as displayed in SmartDashboard.



Virtual Router - Topology

The **Virtual Router Network Configuration** page defines the network topology for the Virtual Router. For an external interface, you define one or more shared external interfaces and a default gateway.



The topology definition consists of the following properties:

- **Interfaces**: Add new interfaces, or modify or delete existing interfaces.
- To add an interface, click **Add**. The **Interface Properties** window opens. Select an interface from the list and define the IP address, net mask and other properties ("Modifying an Interface Definition" on page 63).
- **Routes**: Add network routes between this Virtual Router, Virtual Systems, external network devices and network addresses. Some Warp Link routes are defined automatically and cannot be modified or deleted. You can manually add new routes as well as delete and modify non-Warp Link routes.
- **Add Default Route**: Define the default route as an IP address or Virtual System.
- **Advanced Routing**: Configure source-based routing ("Working with Source-Based Routing" on page 58) rules.

Virtual Router - IPS

Virtual Routers can only run the default protection profile. There are no configurable options here.

Virtual Router - Logs and Masters

The **Logs and Masters** page allows you define logging options for a VSX gateway. The SmartView Monitor ("Tracking Activity with SmartView Monitor" on page 69) section provides additional details for working with logs.

Virtual Router - Advanced

The **Permissions to Install** page allows you to specify user groups whose members are authorized to install policies on this Virtual Router.

Deleting a Virtual Router

You cannot delete a Virtual Router if it is still connected to a Virtual System. Remove all Virtual Router connections before deleting.

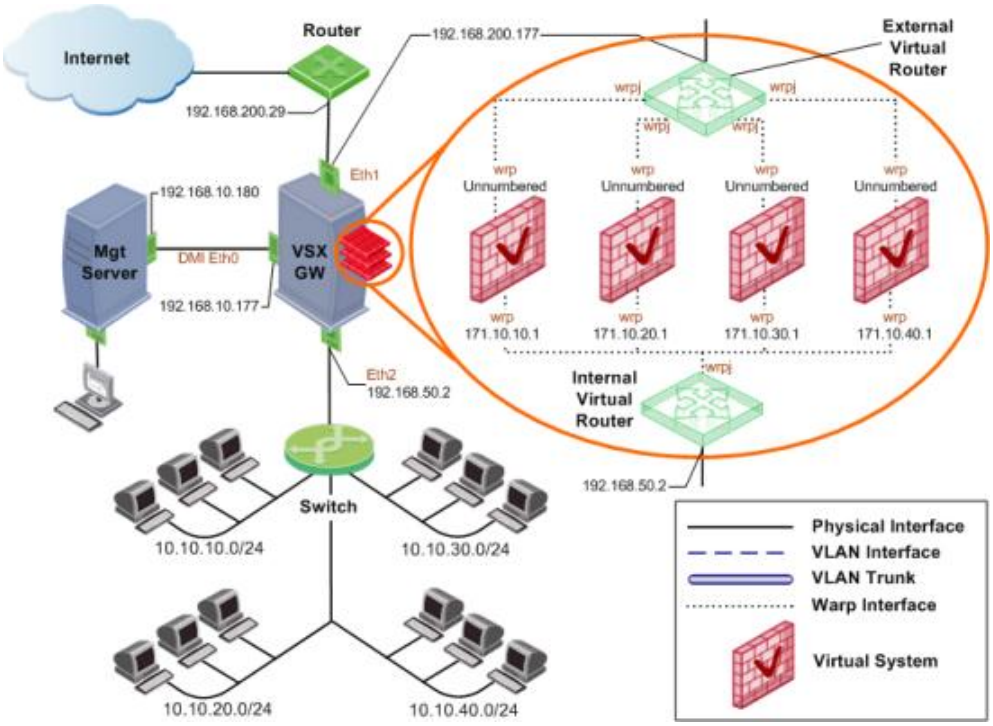
To delete a Virtual Router, right-click the appropriate Virtual Router object on the Object Tree and select **Delete**. Click **Yes** in the confirmation box.

Working with Source-Based Routing

Source-based routing directs traffic to a specific destination based on the source IP address or a combination of the source and destination IP addresses. Rules defining Source-based routing take precedence over ordinary destination-based routing rules.

This section describes how to configure sourced-based routing rules when working in a VSX environment. The procedures for defining source-based rules are the same for Virtual Routers in both VSX gateways and VSX clusters.

Figure 3-18 Source-based routing with Virtual Routers

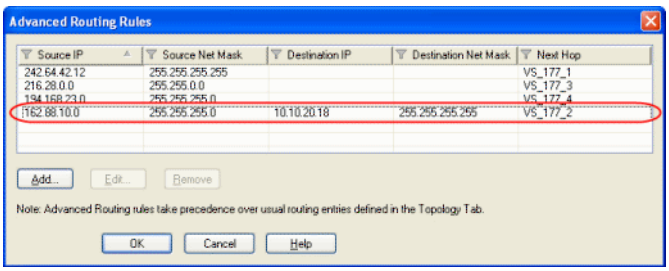


Defining Source-Based Routing Rules

You define Source-based Routing rules from the Topology page in the Virtual Router definition window.

To define source-based routing rules:

1. Open the appropriate internal Virtual Router definition in SmartDashboard and select the **Topology** page. The **Advanced Routing Rules** window opens.



Note that the highlighted rule is based on both a source and destination address, as compared to the preceding rules, which are based on a source address only.

2. Click **Add** to define a new rule or **Edit** to modify an existing rule. The **Add/Edit Route Rule** window appears.

Define the following properties as required:

- **Source IP Address and Net Mask**
- **Destination IP Address and Net Mask (optional)**
- **Next Hop Gateway**: Select a Virtual System from the list.

Working with Dynamic Routing

This section presents procedures for configuring dynamic routing for Virtual Systems and Virtual Routers. Virtual Devices can communicate and distribute routes amongst themselves using dynamic routing protocols.

You configure dynamic routing separately for each Virtual System and/or Virtual Router, each of which has its own dynamic routing daemon and configuration file. You also configure dynamic routing separately on each cluster member.

Enabling Dynamic Routing

To enable dynamic routing:

1. Enable dynamic routing for the appropriate Virtual System or Virtual Router by executing the following commands from the VSX gateway in the expert mode:
   a) `vsx set <vs_id>`, where `<vs_id>` is the device context ID
   b) `drouter enable <vs_id>` (enables dynamic routing)
   c) `drouter start <vs_id>` (starts the dynamic routing daemon)

Configuring Dynamic Routing

To configure dynamic routing:

1. Execute the `router` command.
2. Configure dynamic routing according to your requirements. See the *R75.20 Advanced Routing Suite CLI Reference Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12262) for details.

Disabling Dynamic Routing

To disable dynamic routing:

1. Execute the following commands from the VSX gateway command line:
   a) `drouter stop <vs_id>` (stops the dynamic routing daemon)
   b) `drouter disable <vs_id>` (disables dynamic routing)

Checking Dynamic Routing Status

To check whether dynamic routing is active, execute the `drouter stat <vs_id>` command from the VSX gateway command line. A message appears indicating whether or not dynamic is enabled.

Working with Interface Definitions

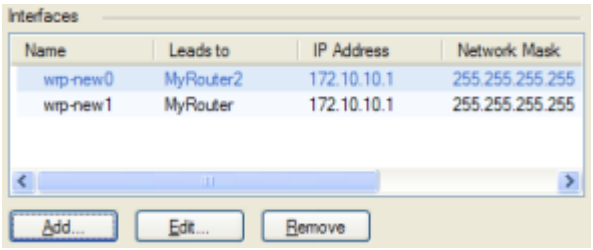
All VSX gateways and Virtual Routers and Virtual Switches contain at least one interface definition. Typically, you define the interfaces during the process of configuring the topology for a given object. Warp interfaces, however, are created automatically based on virtual device definitions and their topology. You cannot modify or delete a Warp interface.

Adding a New Interface

The procedure and options for defining an interface vary according to the object and the network topology. Please note that some properties and pages are not available for certain interface definitions.

To add a new interface:

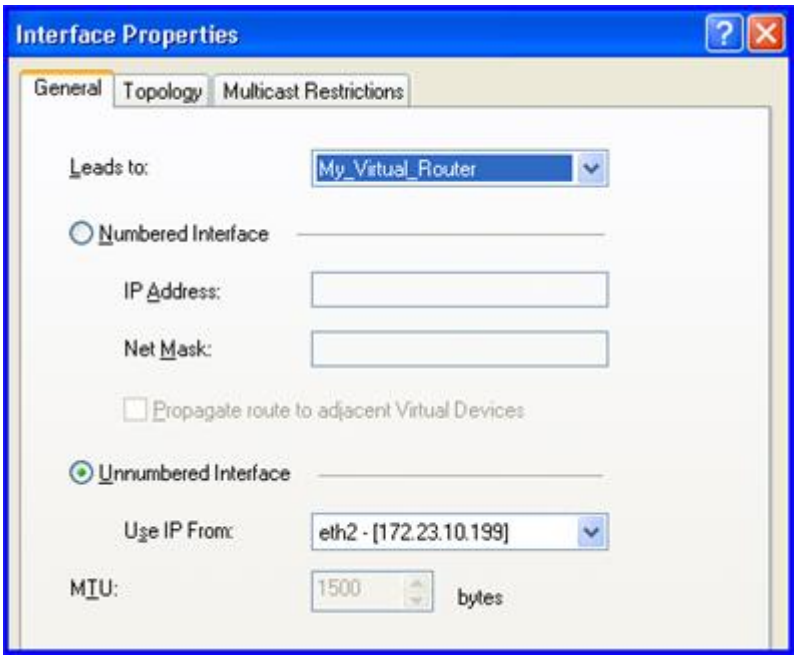
1. On the **Topology** page, click **Add**.
2. If a connection options list appears as shown below, choose the desired connection for this interface.



3. The **Interface Properties** window opens. Configure the appropriate properties as described in the following sections.

Configuring Connection Properties - General

The **General** tab defines the network connections associated with an interface.

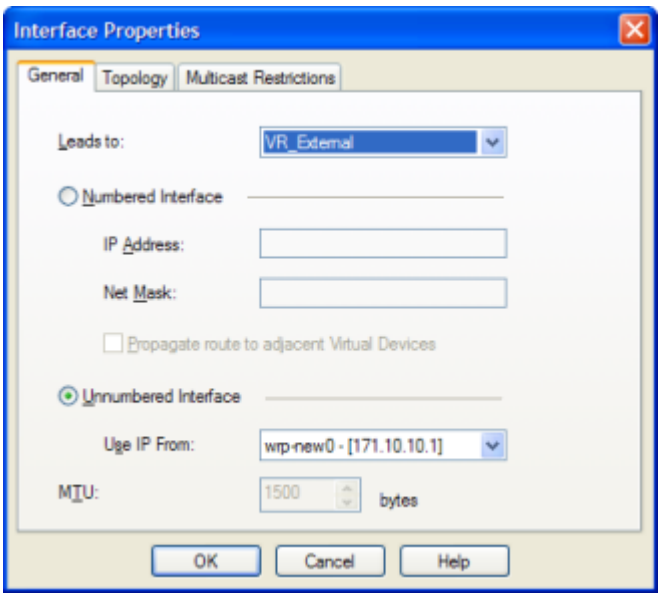


One or more of the following properties may appear depending upon the context.

- **Interface**: Select a physical interface from the list (physical interfaces only).
- **VLAN Tag**: VLAN tag associated with the defined interface.
- **IP Address** and **Net Mask**: IP address and net mask of the device associated with the interface.
- **Propagate route to adjacent Virtual Devices**: Enable to "advertise" the associated device to neighboring devices, thereby enabling connectivity between them. The Route Propagation section (See "Route Propagation" on page 28) provides additional details.
- **MTU**: Maximum transmission unit size in bytes (default = 1,500).

Configuring Connections Leading to Virtual Routers

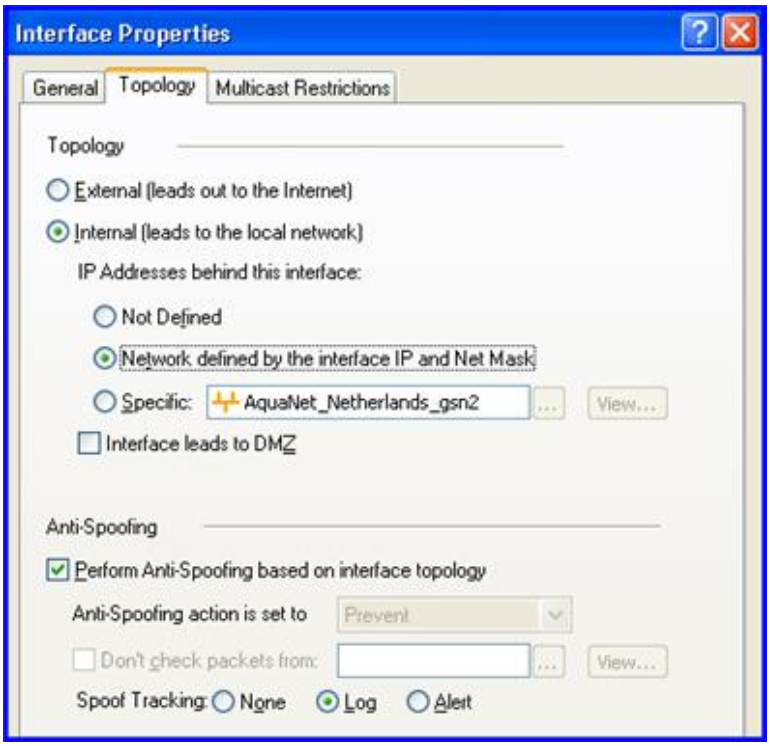
The **General** tab for interface connections leading to Virtual Routers contains connection properties specific to Virtual Routers.



- **Leads to**: Select a Virtual Router from the list.
- **Numbered Interface**: Select this to assign a dedicated Virtual System IP address to an interface leading to a Virtual Router.
- Select a Virtual System address from the list.
- The net mask property is always defined as **255.255.255.255** and cannot be modified.
- **Unnumbered Interface**: Select this option to "borrow" the IP address assigned to another Virtual System interface instead of assigning a dedicated IP address.
- Select the IP from which you wish to "borrow" the IP address from the list.

Configuring Interface Topology

For some interface types, you can directly modify some or all of the following topology properties:



- **External**: The interface leads to external networks or to the Internet.
- **Internal**: The interface leads to internal networks and/or a DMZ and includes the following properties:
  - **Not Defined**: IP routing is not defined for this device.
  - **Network**: Routing is defined by the IP and net mask defined in General Properties.
  - **Specific**: Routing is defined by a specific network or network group.
  - **Interface leads to DMZ**: Defines an interface as leading to a DMZ, which Isolates a vulnerable, externally accessible resource from the rest of an protected, internal network.

Configuring Anti-Spoofing

Attackers can gain access to protected networks by falsifying or "spoofing" a trusted source IP address with high access privileges. It is important to configure anti-spoofing protection for VSX gateways and Virtual Systems, including internal interfaces. You can configure anti-spoofing for an interface, provided that the topology for the interface is properly defined.

If you are using dynamic routing, disable the **Calculate topology automatically based on routing information** option, and manually configure the topology of the Virtual System.

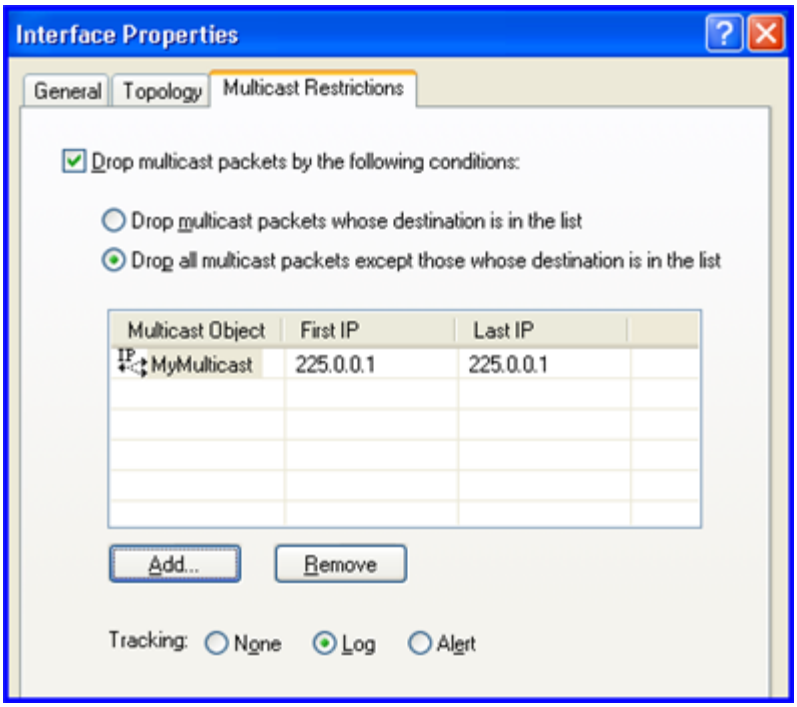
To enable anti-spoofing for an interface, enable the **Perform Anti-Spoofing based on interface topology** option on the **Topology** tab in the **Interface Properties** window. Select a tracking option as appropriate.

Configuring Multicast Restrictions

IP multicasting applications send one copy of each datagram (IP packet) and address it to a group of computers that wish to receive it. Multicast restrictions allow you to define rules that block outbound datagrams from specific multicast groups (IP address ranges). You can define multicast access restrictions for physical and Warp interfaces in a VSX environment.

To enable multicast restrictions:

1. Enable the **Drop multicast packets by the following conditions** option on the **Multicast Restrictions** tab in the **Interface Properties** window.



2. Select one of the following restriction types:
   - **Drop multicast packets whose destination is in the list**
   - **Drop all multicast packets except those whose destination is in the list**
3. Click **Add** to add a multicast address range. The **Multicast Address Range Properties** window opens.
4. Define an **IP address Range** or a **Single IP Address** in the **224.0.0.0** to **239.255.255.255** range.
5. Select a tracking option.

6. Close the window and save the definition.
7. Add a rule to the Rule Base that allows traffic for the specified multicast groups and install the policy.

Modifying an Interface Definition

This sections presents procedures for modifying existing interface definitions and related features.

Selecting and Opening an Existing Interface

Interfaces definitions are always associated with a Virtual Gateway or a Virtual System definition. To work with an existing interface definition:

1. Double click the desired interface in the **Interfaces** section.
2. In the **Interface Properties** window, define the interface properties ("Adding a New Interface" on page 60).

Deleting an Interface

To delete an interface, click **Remove** on the object **Topology** page.

Working with Authentication

Supported Authentication Schemes

Authentication schemes employ user names and passwords to identify valid users. Some schemes are maintained locally, storing user names and passwords on the VSX gateway, while others store authentication information on an external authentication server. Some schemes, such as SecurID, are based on providing a one-time password.

All of the schemes can be used with users defined on an LDAP server. For additional information on configuring a Security Gateway to integrate with an LDAP server, refer to the *SmartDirectory (LDAP)* and *User Management* section in the *R75 Security Management Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11667).

Check Point Password

VSX stores a static password for each user in the management server database. No more software is required.

Operating System Password

VSX can authenticate users by means of a user name and password defined on the management server operating system. You can also use passwords stored in a Windows domain. No additional software is required.

Radius

Remote Authentication Dial In User Service (RADIUS) is an external, server-based authentication protocol that provides authentication services using the UDP protocol.

TACACS, TACACS+

Terminal Access Controller Access Control System (TACACS) is an external, server-based authentication protocol that provides verification services using the TCP protocol. TACACS+ is an enhanced version of the TACACS that supports additional types or authentication requests and response codes.

SecurID

SecurID requires users to possess a token authenticator and to supply a password. Token authenticators generate one-time passwords that are synchronized to an RSA ACE/server. Hardware tokens are key-ring or credit card-sized devices, while software tokens reside on the PC or device from which the user wants to authenticate. All tokens generate a random, one-time use access code that changes approximately every minute. When a user attempts to authenticate to a protected resource, the one-time use code must be validated by the ACE/server.

Configuring RADIUS or TACACS/TACACS+

Two options are available for enabling connectivity between Virtual Systems and external authentication servers:

- **Shared**: Servers are accessible from VSX gateways and clusters
- **Private**: Servers are accessible from Virtual Systems

Shared

When the shared option is configured, all authentication servers are accessible by all Virtual Systems through the VSX gateway. This is the default option.

1. To configure the shared option, use the database tool **GuiDBedit** to set the `shared_external_server` property to TRUE (default setting).
2. The Virtual Systems use the IP address of the VSX gateway. Therefore, connections to external servers have the VSX machine's IP address as their source address, a unique IP address for each cluster member. Virtual Systems on the same cluster member have identical source addresses when accessing the external management server.
3. Verify that the Authentication Server is located on the same network segment as the VSX gateway.
4. Members of the cluster must *not* perform hide NAT on the external server service. To prevent Hide NAT:

   a) On the management server, open the `/opt/CPvsxngxcmp-R67/lib/table.def` file for editing.
5. To the `no_hide_services_ports` table, add the service of the authentication scheme you wish to use. For example UDP 5500 for SecurID. The line should read:

```
no_hide_services_ports = { <500, 17>, <259, 17>, <1701,
17>, <123, 17>, <5500,17> };
```

6. Reinstall the policy on the Virtual System.

Private

When the private option is configured, authentication servers are accessed directly by the Virtual System.

- The Virtual system and the authentication server are located on the same network segment.
- Connections to the external authentication server use the Virtual System's cluster IP address as the source address.
- To configure the private option, use the database tool **GUIDBedit** to set the `shared_external_server` property to FALSE.
- Once the private option has been configured, it is not possible for the Virtual System to connect to other authentication servers in the VSX management network unless an explicit path is created through a Virtual Router or Virtual Switch.
- There is no need to edit the `table.def` file.

Configuring SecurID ACE/Server

There are two options available for enabling connectivity between Virtual Systems and a SecurID ACE/Server:

- **Shared**: Servers are accessible from VSX gateways and clusters

- **Private**: Servers are accessible from Virtual Systems

In both instances, the SecurID ACE/Server sends a shared key (called a "node secret") to its peer ACE/Clients. This key is unique per IP address, and is sent once for each IP address.

Note - Users cannot authenticate to a Virtual System using SecurID when SSL Network Extender and SecureClient are active.

Shared

- To configure the shared option, use the database tool **GUIDBedit** to set the `shared_external_server` property to TRUE.
- Members of the cluster must *not* perform Hide NAT on the external server service.

To prevent Hide NAT:

1. On the management server, open the `/opt/CPvsxngxcmp-R67/lib/table.def` file for editing.
2. Add the UDP 5500 service to the `no_hide_services_ports` table. The line should read as follows:

```
no_hide_services_ports = { <500, 17>, <259, 17>, <1701,
17>, <123, 17>, <5500,17> };
```

3. Reinstall the policy on the Virtual Systems.

   To generate an `sdconf.rec` file, perform the following procedure on the ACE/Server.
4. Generate the `sdconf.rec` file with the IP address of the VSX gateway: VS(0).
5. Copy the `sdconf.rec` file to the appropriate cluster member.

   a) When a Virtual System connects to a VSX gateway at VS(0), place the `sdconf.rec` in the `/var/ace` directory. Create this directory if it does not exist.

   b) In all other cases, place `sdconf.rec` in `$FWDIR/CTX/CTX000X/conf`.

Private

When using the private option for accessing external servers, all the members use the same cluster IP address as the source address for connections to the ACE/Server.

- To configure the private option, use the database tool **GUIDBedit** to set the `shared_external_server` property to FALSE.
- After the first connection that uses SecurID authentication, the ACE Server creates a shared key called `securid`. This "node secret" key is created only once, and sent to the `$FWDIR/CTX/CTX000X/conf` directory of the active cluster member.
- To make this shared key available to the other member gateways, manually copy the "node secret" key from the first gateway. (The ACE/Server will not recreate the key.)

To generate an `sdconf.rec` file, perform the following procedure on the ACE/Server.

1. Generate the `sdconf.rec` file with the cluster IP of the Virtual System.
2. Copy the `sdconf.rec` file to the relevant cluster member.

   a) When a Virtual System connects to a VSX gateway at VS(0), place the sdconf.rec in the `/var/ace` directory. Create this directory if it does not exist.

   b) In all other cases, place `sdconf.rec` in `$FWDIR/CTX/CTX000X/conf`.

For Both Shared and Private Options on SecureID Connections

In order that the active cluster member uses the cluster IP address as part of the hash performed on securID traffic, perform the following steps on all cluster members:

1. Create a file named `/$FWDIR/CTX/CTX<VSID>/conf/sdopts.rec`
2. Enter the client IP address:

```
CLIENT_IP=<Virtual System cluster IP>
```

3. Perform `cpstop/cpstart`.
   - Perform the following procedure on all cluster members:
4. Open the `/etc/services` file for editing.
5. Add the following lines:

```
securid 5500/udp
securidprop 5510/tcp
```

The Effect of Upgrading on Authentication Processes

- An existing Virtual System that has been upgraded to the current version, receives the default settings for authentication with external servers.
- If the Virtual System was originally created on a management server located on the same network segment as the external authentication server, connectivity may be lost until the "private" option is enabled.

Client/Session Authentication

VSX supports the following client/session authentication schemes:

- Client authentication over TELNET (on port 259)
- Client authentication over HTTP/HTTPS (on port 900)

For a complete description of these features, see the *R75 IPS Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11663).

VSX Limitations

- User authentication is not supported
- The following client authentication methods are not supported in VSX environments:
  - Partially automatic
  - Fully automatic
  - Single Sign-on (UserAuthority)

Configuring Client/Session Authentication

In a VSX environment, you configure Client/Session authentication settings by manually editing the *$FWDIR/conf/cpauthd.conf* file, located on the VSX Gateway.

Note - This procedure differs configuring client/session authentication for physical security gateways.

You must configure client/session for the VSX Gateway. These settings apply, by default, to all Virtual Systems located on the gateway.

You can optionally configure client/session authentication for specific Virtual Systems. Virtual System specific settings override the default settings for that Virtual System only. Virtual Systems that do not have their own settings inherit the default settings.

Configuring Authentication for the VSX Gateway

To configure client/session authentication for the VSX Gateway:

1. Backup *$FWDIR/conf/cpauthd.conf* .

2. Open *$FWDIR/conf/cpauthd.conf* , on the VSX Gateway machine using a text editor.
3. Add or modify the following attributes according to the table:

Attribute	Default Value	Explanation
clauth_port	259	The TCP port on which client authentication over TELNET is done. 0 = Client authentication over TELNET is disabled.
clauth_http_port	900	The TCP port on which client authentication over HTTP/HTTPS is done. 0 = Client authentication over HTTP/HTTPS is disabled.
clauth_http_ssl	0	0 = HTTPS client authentication is disabled. 1 = HTTPS client authentication is enabled.
clauth_http_nickname	none	Specifies the certificate nickname when client authentication is performed over HTTPS. This attribute must match the virtual system certificate nickname as configured using SmartDashboard (**Virtual System >VPN >Certificate List**).

4. Run *cpwd_admin stop -name FWD -path "$FWDIR/bin/fw" -command "fw kill fwd"*.
5. Run *cpwd_admin start -name FWD -path "$FWDIR/bin/fwd" -command "fwd"*.

Configuring Authentication for Specific Virtual Systems

To configure client/session authentication for the VSX Gateway:

1. Backup *$FWDIR/CTX/CTX#/conf/cpauthd.conf*, where CTX# refers to the specific Virtual System directory.
2. Delete the original *$FWDIR/CTX/CTX#/conf/cpauthd.conf*.
3. Open *$FWDIR/conf/cpauthd.conf* to *FWDIR/CTX/CTX#/conf/cpauthd.conf* using a text editor.
4. Add or modify the following attributes according to the table:

Attribute	Default Value	Explanation
clauth_port	259	The TCP port on which client authentication over TELNET is performed. 0 = Client authentication over TELNET is disabled.
clauth_http_port	900	The TCP port on which client authentication over HTTP/HTTPS is performed. 0 = Client authentication over HTTP/HTTPS is disabled.
clauth_http_ssl	0	0 = HTTPS client authentication is disabled. 1 = HTTPS client authentication is enabled.
clauth_http_nickname	none	Specifies the certificate nickname when client authentication is performed over HTTPS. This attribute must match the virtual system certificate nickname as configured using SmartDashboard (**Virtual System >VPN >Certificate List**).

5. Run *cpwd_admin stop -name FWD -path "$FWDIR/bin/fw" -command "fw kill fwd"*.
6. Run *cpwd_admin start -name FWD -path "$FWDIR/bin/fwd" -command "fwd"*.

Notes

1. cpauthd.conf is used instead of $FWDIR/conf/fwauthd.conf on a non-VSX Gateway.
2. All virtual systems (other than the default Virtual System) are assigned a symbolic link in the *$FWDIR/CTX/CTX#/conf/cpauthd.con* file. This link points to *$FWDIR/conf/cpauthd.conf*, where CTX# refers to the specific Virtual System directory.

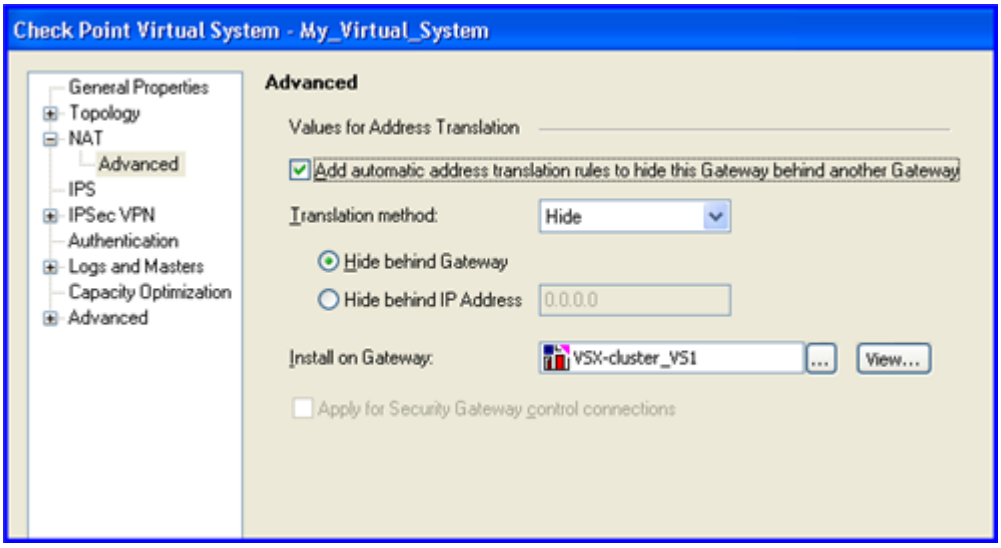
Working with Network Address Translation

This section describes the process for using Network Address Translation (NAT) in a VSX deployment. The procedures described in this section assume that the reader is familiar with NAT concepts and their implementation in Check Point products. For more about NAT, see the *Network Address Translation* chapter in the *R75.20 Firewall Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12267).

VSX supports NAT for Virtual Systems much in the same manner as a physical firewall. When a NAT enabled (Static or Hide) Virtual System connects to a Virtual Router, the translated routes are automatically forwarded to the appropriate Virtual Router.

Configuring NAT

You configure NAT using the **NAT** page in the **Virtual System** window. Hide or Static NAT addresses configured in this manner are automatically forwarded to the Virtual Router to which the Virtual System is connected. Alternatively, you can manually add NAT routes on the **Topology** page in the **Virtual Router** window.



To enable and configure NAT for a Virtual System:

1. Enable the **Add Automatic Address Translation** option.
2. Select a translation method from the list.
   - **Hide NAT**: Hide NAT only allows connections originating from the internal network. Internal hosts can access internal destinations, the Internet and other external networks. External sources cannot initiate a connection to internal network addresses.
   - **Static NAT**: Static NAT translates each private address to a corresponding public address.
3. If you select **Hide NAT**, select one of the following options:
   - **Hide behind Gateway** hides the real address behind the VSX gateway external interface address. This is equivalent to hiding behind the address 0.0.0.0, or
   - **Hide behind IP Address** hides the real address behind a virtual IP address, which is a routable, public IP address that does not belongs to any real machine.
4. If you selected **Static NAT**, enter the static IP address in the appropriate field.
5. Select the desired VSX gateway from the **Install on Gateway** list.

Tracking Activity with SmartView Monitor

SmartView Monitor is the Graphical User Interface application from which all gateway and Virtual Systems/Routers statuses are displayed. SmartView Monitor displays a snapshot of installed Check Point products including VSX.

For more information on using SmartView Monitor, refer to the *R75 SmartView Monitor Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11672).

SmartView Monitor displays each VSX gateway or cluster as a regular Check Point gateway, but with a different icon. For each VSX gateway or cluster, the VSX gateway object shows OS level information as well as CPU Memory and Disk Usage information.

SmartView Monitor connects to and validates each Virtual System as an independent gateway. If one Virtual System is down, this information will be reflected in SmartView Monitor even though the other Virtual Systems on the VSX gateway/cluster are functioning normally.

Chapter 4

Using VSX with Multi-Domain Security Management

You can manage a VSX deployment using Multi-Domain Security Management. This chapter assumes that you are familiar with the Multi-Domain Security Management product. Only procedures specific to VSX deployments are discussed. See the *R75 Multi-Domain Security Management Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11683).

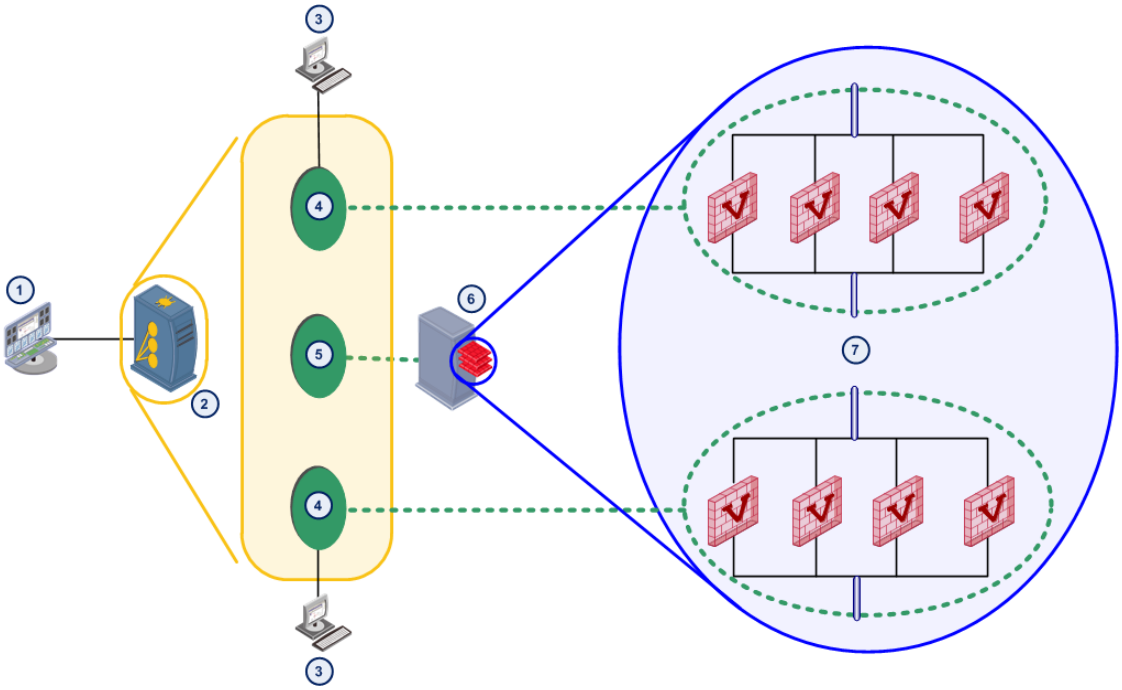
In This Chapter

Overview	70
Licensing VSX with Multi-Domain Security Management	71
VSX Provisioning	74
Defining Multi-Domain Servers	74
Defining Domains and Servers	76
Working with Virtual Devices	81

Overview

Check Point Multi-Domain Security Management is a centralized security management solution that addresses the unique requirements of service providers and large enterprises. By using Multi-Domain Security Management, administrators can centrally manage multiple independent networks, often belonging to different Domains, divisions, or branches.

Figure 4-19 Multi-Domain Security Management Managing VSX



	Description

	Description
1	SmartDomain Manager
2	Multi-Domain Server
3	SmartDashboard
4	Domain Management Server
5	Main Domain Management Server
6	VSX Gateway
7	VSX Virtual System in Domain Management Servers

The **Multi-Domain Server** is a central management server that hosts the network management and security policy databases for these networks. Each independent domain is represented by a **Domain**, which provides the full functionality of a Security Gateway. Each Domain Management Server can host Virtual Systems, Virtual Routers and Virtual Switches as well as physical Check Point gateways. We recommend that you manage each VSX gateway with its own Domain Management Server.

The Domain Management Server that manages a VSX gateway or cluster is known as a **Main Domain Management Server**. You can host multiple gateways and/or clusters on one Multi-Domain Server. We recommend that you manage each gateway and cluster with its own main Domain Management Server. Virtual Systems belonging to a given Domain can be distributed among multiple VSX gateways and clusters.

The **SmartDomain Manager** is a centralized management solution for Domains, Domain Management Servers and the Multi-Domain Security Management environment. Each Domain Management Server uses its own instance of SmartDashboard, which is accessible only via the SmartDomain Manager, to provision its Virtual Devices and physical gateways, as well as to manage their security policies.

Licensing VSX with Multi-Domain Security Management

Check Point software is activated with a license key. To obtain a license key, register the certificate key (that appears on the back of the software media pack) with the Check Point User Center. The certificate key is used to generate a license key for the products that you are either evaluating or purchasing.

To purchase the required Check Point products, contact your reseller. Check Point software that has not yet been purchased functions for 15 days only.

Multi-Domain Security Management Licenses

Multi-Domain Security Management licenses are associated with the IP address of the licensed entity. Multi-Domain Server licenses are based on the Multi-Domain Server type.

- **Multi-Domain Server -** This license covers the administrator access point to the Multi-Domain Security Management environment and is bound to the Multi-Domain Server IP address. The SmartDomain Manager can only connect to Multi-Domain Servers machines with a valid license.
- **Domain Management Server License:** Each individual Domain Management Server requires its own license, bound to its IP address. Domain Management Server licenses cover a predefined number of enforcement points (Virtual Systems and/or physical Check Point gateways). Domain Management Server licenses are available for quantities of one, two or four enforcement points as well as an unlimited license covering an unlimited number of enforcement points.

  Individual Domain Management Server licenses are recommended for mixed networks consisting of both Virtual Systems and physical Check Point gateways. V
- **Domain Management Server Pro**: Domain Management Server Pro licenses, which enable additional management features at the Domain Management Server level, can be purchased in bulk and are called Pro Add-ons for Multi-Domain Server.
- **Multi-Domain Server**: This is a comprehensive license that enables real-time logging, tracking and log management for a predefined number of Log Servers hosted on a dedicated Multi-Domain Log Server.

Multi-Domain Log Server licenses are available in packs of 10, 25, 50, 100 and 250. This license is bound to the Multi-Domain Log Server IP address.

- **Domain Log Server License:** This license is intended for a single Domain with log files hosted only on the Multi-Domain Server. The license is bound to the Log Servers IP address. Individual Log Servers licenses are **not** required for Log Servers hosted on an Multi-Domain Log Server.

You can import Multi-Domain Security Management licenses can using the Check Point command line licensing tool or the SmartDomain Manager. See the *Multi-Domain Security Management Administration Guide*.

VSX Gateway/Cluster Member Licenses

Each VSX gateway or cluster member requires its own license, bound to the gateway or cluster member IP address. Each gateway/cluster license covers a predefined number of Virtual Systems (10, 25, 50, 100 and 250) and these licenses are cumulative.

VSX/Domain Management Server Bundle Licenses

VSX/Domain Management Server bundle licenses are a cost-effective solution for purchasing Domain Management Server licenses that allow you to manage a combination of multiple Domain Management Servers and multiple Virtual Systems. Bundle licenses cover a predefined number of Domain Management Servers and Virtual Systems and are available in packs of 10, 25, 50, 100 and 250.

Each license bundle allows you to manage the following items, according to its limit:

- One Main Domain Management Server
- Up to (10, 25, 50, 100 or 250) Domain Domain Management Servers
- Up to (10, 25, 50, 100 or 250) Virtual Systems

A bundle license allows you to manage any combination of Domain Management Servers and Virtual Systems, providing that the total does not exceed the license limit. Each Domain Management Server can host any number of Virtual Systems up to the license limit. For example, you can define any of the following combinations with one 10 pack bundle license:

- One main Domain Management Server and 10 Domain Domain Management Servers containing one Virtual System each
- One main Domain Management Server and one Domain Domain Management Server containing 10 Virtual systems
- One main Domain Management Server, one Domain Management Server containing four Virtual Systems and three Domain Management Servers, each containing two Virtual Systems

In addition to the VSX/Domain Management Server bundle license, you must purchase and install Multi-Domain Server licenses for each Multi-Domain Server and VSX gateway/cluster licenses for the required number of Virtual Systems. Furthermore, if you are using Domain logging, you must purchase the appropriate licenses.

Note - Virtual Routers and Virtual Switches do not require licenses. There is no limit to the number that you can define.

Bundle License Packs are Cumulative

The maximum number of Domain Management Servers and Virtual Systems covered by your license is equal to the sum of all bundle license packs installed. For example, if you purchase two 10 license packs and one 50 license pack, you can define up to three main Domain Management Servers, 70 Domain Domain Management Servers and 70 Virtual Systems. As with individual license packs, you can distribute the Virtual Systems amongst Domain Management Servers so long as the cumulative total of Domain Management Servers and Virtual Systems does not exceed the maximum.

Limitations of VSX/Domain Management Server Bundle Licenses

- Bundle licenses only cover Virtual Systems. If you wish to mix physical devices (Security Gateways, etc.) and Virtual Systems on one Domain Management Server, you must also add the appropriate regular Domain Management Server licenses.
- Virtual Systems installed on a mixed Domain Management Server do not use any license "slots" from a regular license. Similarly, physical devices do not use Virtual System license "slots".
- Bundle licenses cannot be installed on a system using Multi-Domain Security Management Enterprise Edition Domain Management Servers.

High Availability Licenses

Multi-Domain Security Management supports management high availability. By installing a High Availability VSX-Domain Management Server bundle license on a second Multi-Domain Server machine, the specified number of Virtual Systems can be hosted by secondary Domain Management Servers. The capabilities and limitations of this license are similar to those of the regular VSX/Domain Management Server bundle.

Upgrading Licenses

The license upgrade procedure can be performed if you have purchased any of the Enterprise Software Subscription services.

License upgrade fails for products and accounts for which you do not have software subscriptions. To manage your accounts, licenses, and Enterprise Support Programs coverage (under Support Programs), log in to: http://usercenter.checkpoint.com. License upgrade is performed by means of a tool that automatically upgrades both locally and centrally managed licenses.

Using the tool, you can upgrade all licenses in the entire managed system. License upgrade can also be performed manually, per license, in the User Center. For instructions, refer to the Administration Guide in the User Center at: https://usercenter.checkpoint.com/pub/usercenter/faq_us.html.

For the latest information and downloads regarding License upgrades, refer to:

http://www.checkpoint.com/downloads/quicklinks/utilities/ngx/license_upgrade.html

The Trial Period

When installing a Check Point product for the first time, users receive a 15 day trial period, during which the product is fully functional. Once the trial period expires, you must purchase and install the appropriate permanent product licenses. During the trial period, you can define up to 200 Domain Management Servers and up to five Virtual Systems per Domain Management Server.

To install a VSX bundle license before the trial license expires, you must first disable the trial license. To do so, execute the following command: `cpstop_util CPSTOP_SepTiCEmias 1`.

License Violations

In the event of a license violation, the following may occur:

- The system shows an error message
- Depending on the type of license violation, you may not be able to connect to the Multi-Domain Server using the SmartDomain Manager
- Alerts appear in the SmartDomain Manager. (An indication regarding the nature of the license violation also appears on the SmartDomain Manager status bar)
- Audit entries are generated in SmartView Tracker stating the nature of the violation
- If an Multi-Domain Server is in a license violation state, you cannot define new Domain Management Servers, VSX gateways, clusters, or virtual devices

For More Information

For more information regarding licensing, refer to the Check Point User Center (http://usercenter.checkpoint.com).

VSX Provisioning

The procedures for provisioning and configuring VSX gateways, clusters and virtual devices using the Multi-Domain Security Management model are essentially the same as described for the Security Gateway management model. The principle difference is that you must first create and configure each Domain and its associated Domain Management Server objects using the SmartDomain Manager.

Each individual Domain Management Server is functionally equivalent to one Security Gateway. It has its own SmartDashboard instance that you use to provision, configure and manage network objects and security policies.

The steps for provisioning a VSX environment in using the Multi-Domain Security Management model are as follows:

1. Define and configure Multi-Domain Server ("Defining an Additional Multi-Domain Server in the SmartDomain Manager" on page 75) and Multi-Domain Log Server as appropriate for your deployment.
2. Create and configure a Domain ("Creating a New Domain Object" on page 76) and a main Domain Management Server for each VSX gateway and/or VSX cluster using the SmartDomain Manager.
3. Create and configure VSX gateway ("Creating a New VSX Gateway" on page 32) and/or cluster objects ("Creating a New Cluster" on page 93) using the main Domain Management Server SmartDashboard. Modify the default security policy for these objects if desired.
4. Define individual Domains and Domain Management Servers ("Creating a New Domain Object" on page 76) as required for your deployment.
5. Create and configure Virtual Systems ("Creating a New Virtual System" on page 42) and other virtual devices for each Domain using that Domain's SmartDashboard.

Defining Multi-Domain Servers

This section briefly presents the procedures for installing and deploying Multi-Domain Server machines in a VSX/Multi-Domain Security Management environment. For complete Multi-Domain Server installation and definition processes, see the *Installation Guide (*http://supportcontent.checkpoint.com/documentation_download?ID=10327*)* and the *Multi-Domain Security Management Administration Guide (*http://supportcontent.checkpoint.com/documentation_download?ID=8741*).*

When working with management High Availability, you define at least two Multi-Domain Server machines. You can also employ multiple Multi-Domain Server machines to efficiently distribute management traffic (management Load Sharing) by creating multiple Domain Management Servers for individual Domains. For Load Sharing, define a Domain Management Server for each Multi-Domain Server.

Installing a New Multi-Domain Server

To install and initially configure a new Multi-Domain Server, perform the procedures as summarized below. For detailed procedures and explanations of the various options, please refer the *High End Security Product Suite Getting Started* Administration Guide.

1. Install the Multi-Domain Server using the installation CD or installation file downloaded from the Check Point Download Center. Make certain that you have superuser permissions.
   a) On SecurePlatform, the installation routine runs automatically.
   b) On Solaria and Linux platforms, navigate to the appropriate directory and execute the `mds_setup` script.
2. Follow the instructions on the screen.
3. After the rebooting computer, enter the `sysconfig` command to perform the initial configuration. Follow the instructions on the screen.

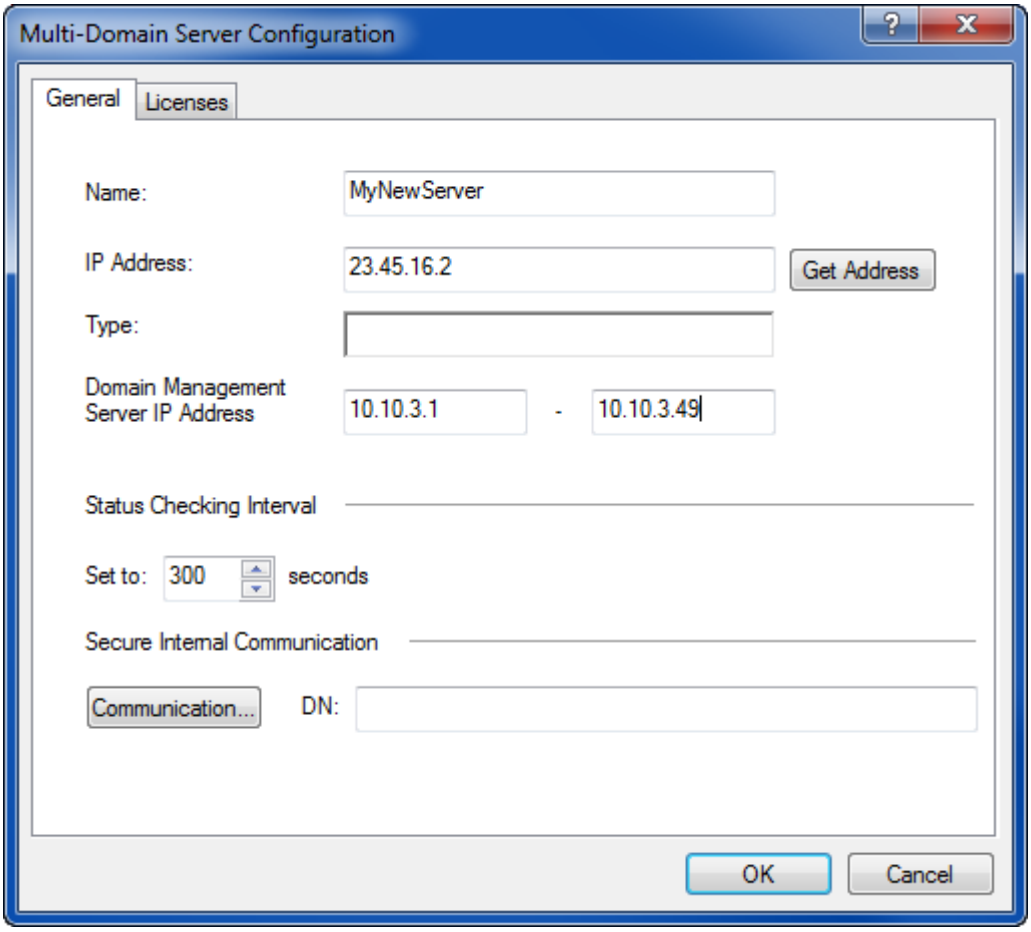
Note - You must always define the first Multi-Domain Server as the primary Multi-Domain Server. Each additional Multi-Domain Servermust be defined as a secondary Multi-Domain Server. To define a Multi-Domain Server as a secondary station, respond '**no**' to the prompt "`Are you installing the Primary Multi-Domain Server [yes/no]?`" during the initial configuration process. Once you complete the initial configuration, you cannot change this definition.

4. Run the `mdsstart` command to start the Multi-Domain Server.
5. Install the VSX management add-on for Security Management/Multi-Domain Security Management as described in the *High End Security Product Suite Getting Started* Administration Guide.
6. Run the SmartDomain Manager and log in. The primary SmartDomain Manager and previously defined SmartDomain Managers appear.

Defining an Additional Multi-Domain Server in the SmartDomain Manager

To add another Multi-Domain Server to your deployment:

1. In **Multi-Domain Server Contents** view, right click the Multi-Domain Security Management line and select **Add New Multi Domain Server** from the option menu. The **Multi Domain Server Configuration** window appears.



2. Provide the following information on the **General** tab as required:
   - **Multi-Domain Server Name**: Text string containing only alphanumeric characters and the underscore character
   - **IP Address:** Routable IP address for the Multi-Domain Server management interface
   - **Domain Management Server IP Address Range**: Optionally specify the first and last IP addresses in the range from which Multi-Domain Security Management assigns virtual IP address for newly defined Domain Management Servers. These addresses must be routable and unique (i.e. not used on other Multi-Domain Server machines).

- **Status Checking Interval**: Interval in seconds between Multi-Domain Server/Domain Management Server status checks (Default = 300).
- **Secure Internal Communication Trust**: Click Communication to open the

3. On the Licenses tab, add Multi-Domain Server licenses as required for your deployment. Click **Add** to add a new license or **Fetch From File** to import a license from a file.
4. Click **OK** to continue. A message appears recommending that you initialize synchronization between Multi-Domain Server machines at this time. Click **Yes** to continue. When the initialization process finishes, the Multi-Domain Server appears as **standby** in the Multi-Domain Server contents view.
5. To fully synchronize the new Multi-Domain Server, right click on the Multi-Domain Server in the **High Availability - Multi-Domain Server Contents** view and select **Synchronize** from the option menu.

Defining Domains and Servers

This section describes define and manage Domains using the SmartDomain Manager. A **Domain** is a Multi-Domain Security Management object that represents a discrete business entity whose networks are protected by Check Point security solutions, including VSX. In many Multi-Domain Security Management deployments, Domain objects represent the physical Domains of a managed service provider or data center.

The **Domain** is a Multi-Domain Security Management object that manages a protected network for a specific Domain, emulating the functionality of Security Gateway. The Domain Management Server manages all Virtual Systems and other virtual devices, as well as any physical Security Gateways and other Check Point devices protecting the Domain network. An individual Domain can have multiple Domain Management Servers, but they must reside on separate Multi-Domain Server.

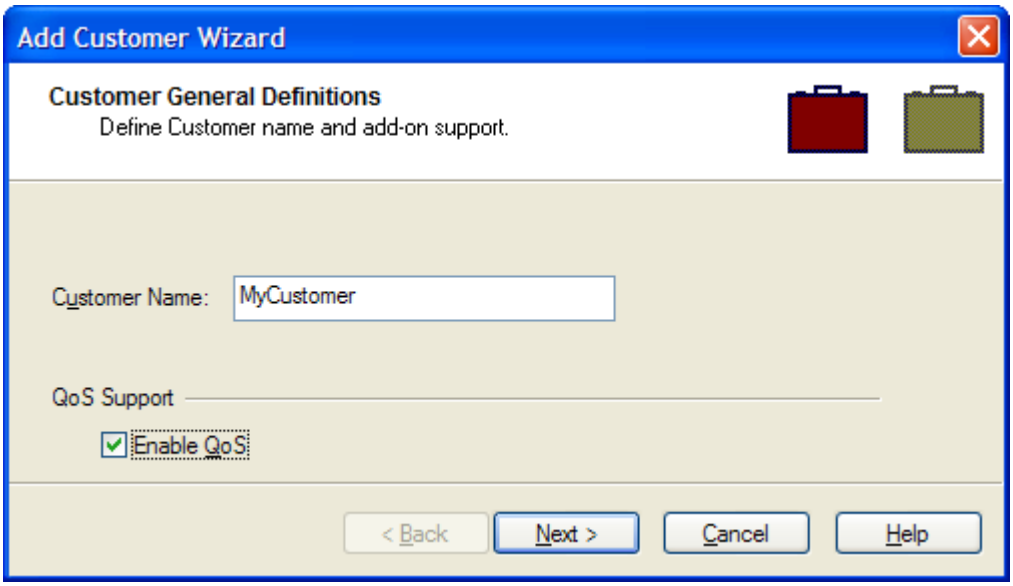
Once you have defined Domain Management Servers for a Domain, you can begin defining Virtual Systems, other virtual devices and physical devices for your Domain. Proceed to the Working with Virtual Devices procedure ("Working with Virtual Devices" on page 81) to define virtual devices for a Domain Management Server.

Creating a New Domain Object

To create a new Multi-Domain Security Management Domain, perform the following procedures:

Starting the Add Domain Wizard

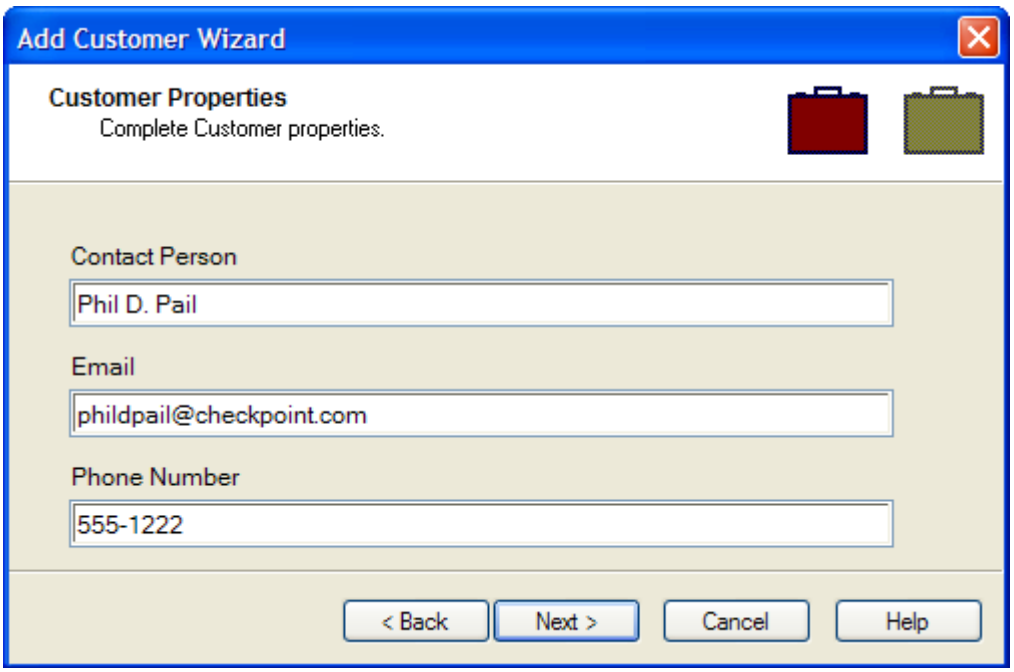
1. In the **Domain Contents** area of the SmartDomain Manager, right-click on **Multi-Domain Security Management** and select **New Domain**. The **Add Domain** wizard opens, displaying the **General Definitions** page.



2. Enter an object name for the Domain in the Domain Name field. Only alphanumeric characters and the underscore character are valid.
3. If you are defining a Domain Management Server that will only manage Virtual Systems, clear the **Enable QoS** option, as this option is not relevant for VSX environments.

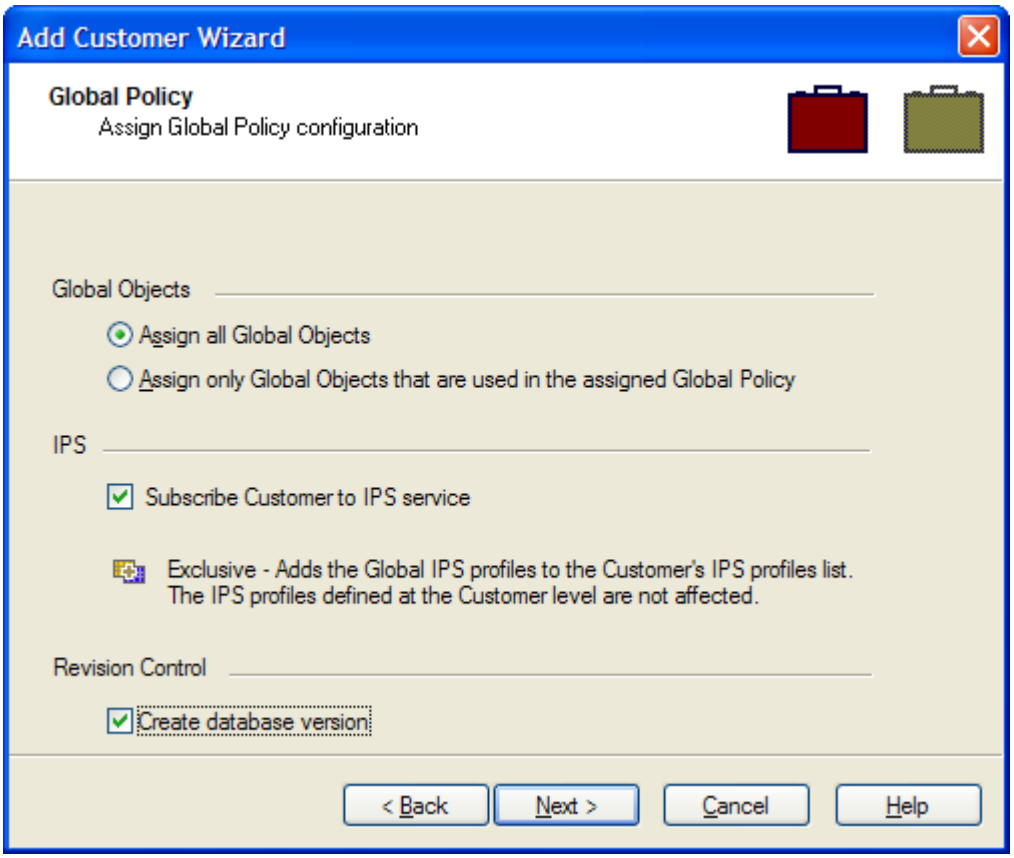
Domain Properties Page

Enter the Domain or business entity name, contact person name and contact person email in the appropriate fields on the Domain Properties page:



Global Policy Page

The **Global Policy** page defines how global policies are applied to a Domain Management Server. **Global Policies** are collections of rules and global objects that apply by default to the entire Multi-Domain Security Management environment or to defined groups of Domains.


Add Customer Wizard
Global Policy
Assign Global Policy configuration
Global Objects
Assign all Global Objects
Assign only Global Objects that are used in the assigned Global Policy
IPS
Subscribe Customer to IPS service
Exclusive - Adds the Global IPS profiles to the Customer's IPS profiles list.
The IPS profiles defined at the Customer level are not affected.
Revision Control
Create database version
< Back
Next >
Cancel
Help

Global Objects

When assigning a global policy to a Domain Management Server, you can choose to assign all global objects or to assign only those objects required by the global policy Rule Base. Select the appropriate option for this new Domain.

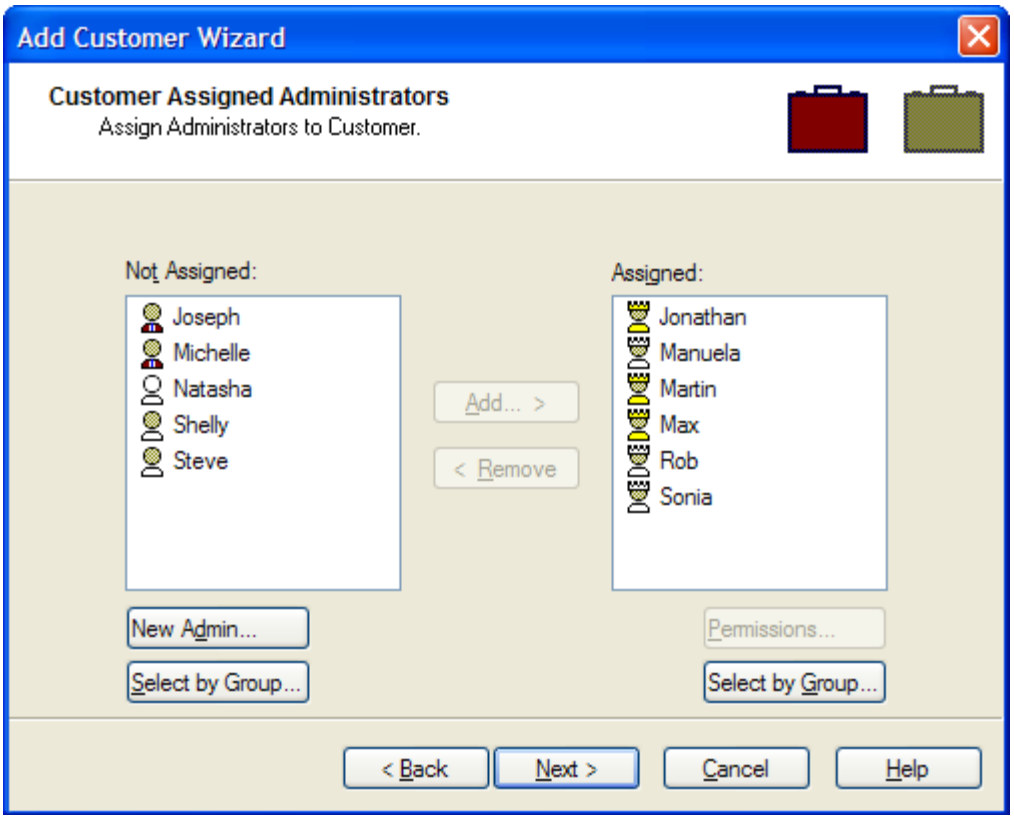
IPS

Enable the **Subscribe Domain to IPS Service** option to assign the global IPS policy to this Domain along with the global policy. Choose one of the following global IPS policy options:

- **Override**: Replace all IPS configuration settings with the global settings (default)
- **Merge**: Preserve any manual IPS configuration settings for this Domain, but update all other settings using the latest Global settings

Domain Assigned Administrators Page

This page allows you to assign and define specific administrators that are authorized to manage this Domain and his protected networks. This is in addition to the Multi-Domain Security Management superusers who are assigned automatically.



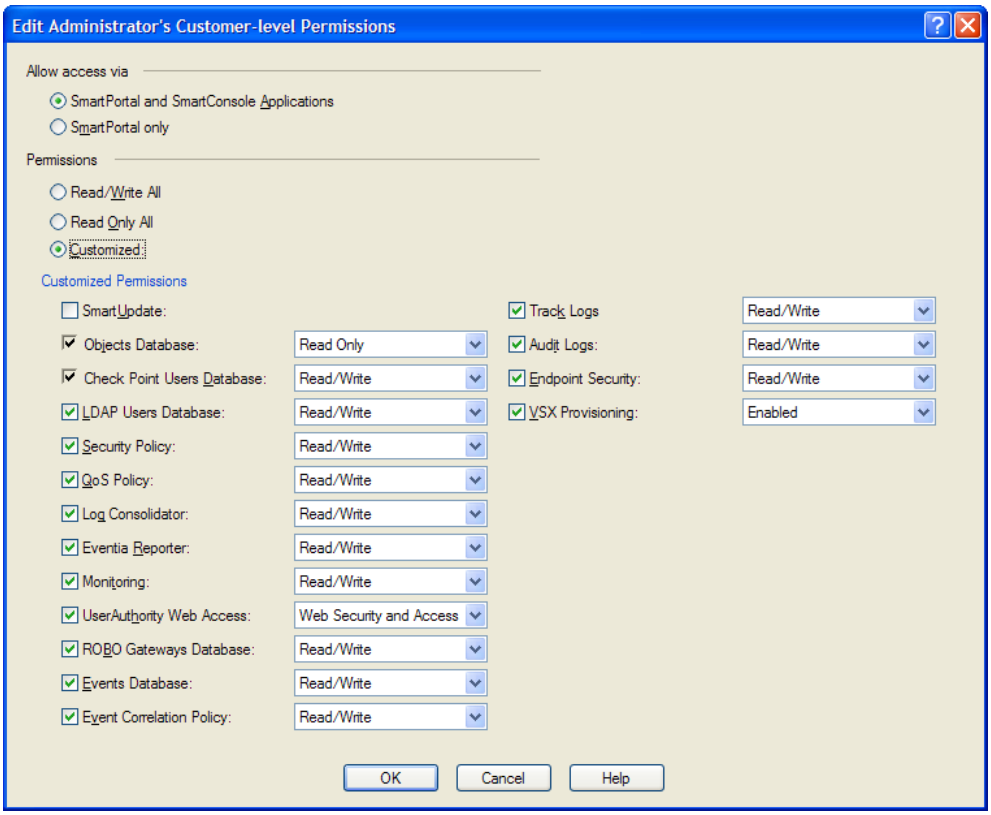
- To assign an administrator to this Domain, select one or more administrators or administrator groups from the **Not Assigned** column and click **Add**. The administrator moves to the **Assigned** column and the **Edit Administrator's Domain-Level Permissions** window opens. Assign administrator permissions ("Assigning Permissions to Administrators" on page 78).
- To remove an administrator, select one or more administrators or administrator groups from the **Assigned** column and click **Remove**. The administrator moves to the **Not Assigned** column.
- To create a new administrator, click **New Admin** and configure the appropriate properties. For more information on assigning/un-assigning or adding new administrators, refer to the *Multi-Domain Security Management Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=8741).

Assigning Permissions to Administrators

You can define custom permissions for individual administrators and administrator groups that allow or deny access to specific administrative tasks. Of particular interest to VSX users, is the option to restrict an administrator's permission to provision VSX objects.

To assign permissions:

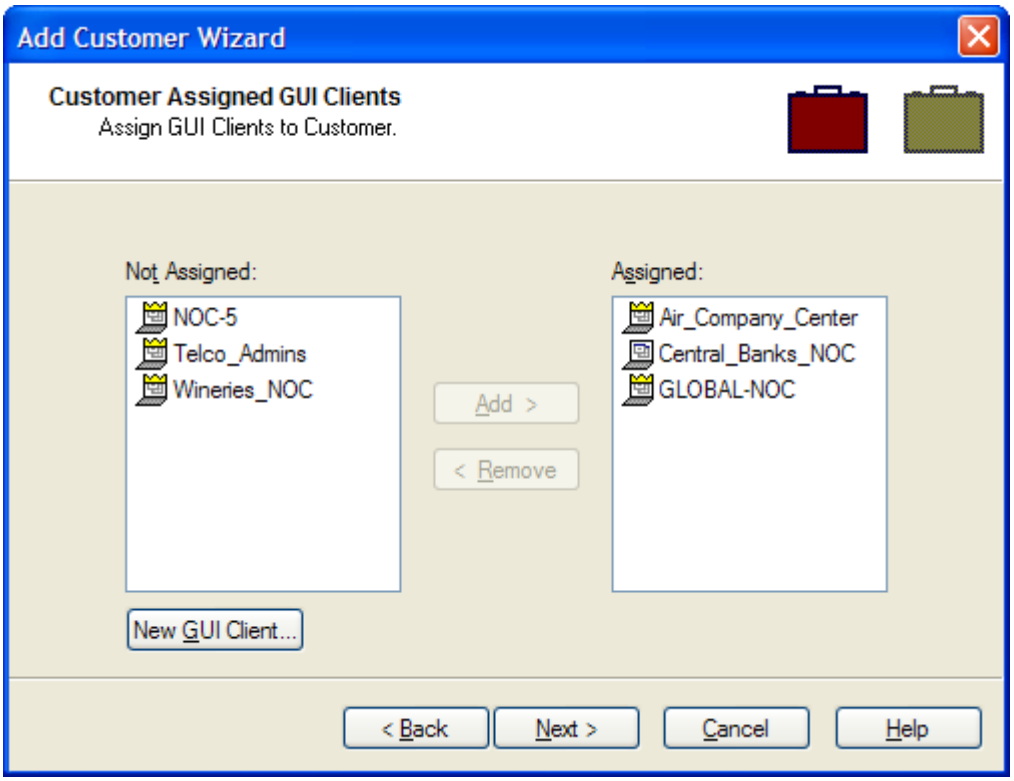
1. Select an administrator in the **Not Assigned** column and click **Add** or select an administrator in the **Assigned** window and click **Permissions**. The **Edit Administrator's Domain-Level Permissions** window opens.



2. Select the **Customized** option to customize administrator permissions, and enable or disable individual task permissions as appropriate. Refer to the online help for details regarding individual options and task permissions.

Domain Assigned GUI Clients Page

This page allows you to assign and create GUI clients that are authorized work with this Domain.



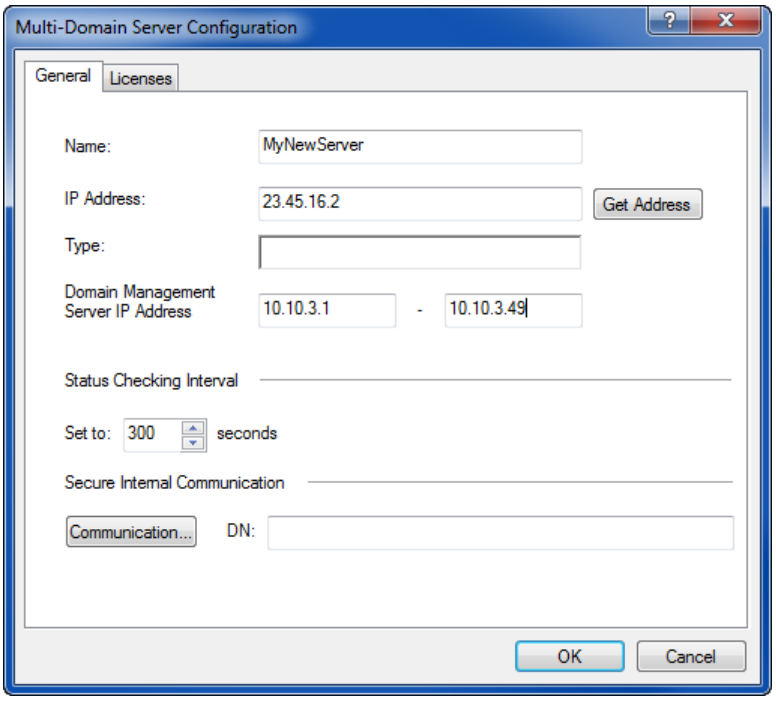
- To assign an predefined GUI client to this Domain, select one or more GUI clients or groups from the **Not Assigned** column and click **Add**. The GUI client moves to the **Assigned** column.
- To remove an GUI client, select one or more GUI clients or groups from the **Assigned** column and click **Remove**. The administrator moves to the **Not Assigned** column.
- To define a new GUI client, click **New GUI Client** and configure the appropriate properties. For more information, refer to the *Multi-Domain Security Management Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=8741).

Adding a Domain Management Server

The First Domain Management Server let you to define a Domain Management Server to manage this Domain's network. Select **Yes** to continue with the Domain Management Server definition or **No** to define the Domain Management Server later. If you select Yes, you can define up to two Domain Management Servers for this Domain at this point. You can add additional Domain Management Servers later.

Note - You can only add one Domain Management Server for per Domain on each Multi-Domain Server defined in your system. For example, you must have at least three Multi-Domain Servers if you wish to define three Domain Management Servers for a Domain. Additional Domain Management Servers can be used as high availability standbys or to spread management traffic over several Multi-Domain Server machines.

To add a Domain Management Server after you complete a Domain definition, simply right click on the Domain in the **Domain Contents** view and select **Add Domain** from the option menu.



To create a Domain Management Server, enter the following information:

- **Name**: Domain Management Server object name for the Domain in the Domain Name field. Only alphanumeric characters and the underscore character are valid.
- **Multi Domain Server**: Select an Multi-Domain Server from the list
- **IP Address**: Virtual IP assigned to this MDA. Click **Get Address** to resolve an address by the Domain Management Server name or to automatically assign an address from a predefined pool of addresses. You can also manually enter an IP address.
- **Licenses**. Attach a Domain Management Server license. If you are using a VSX/Domain Management Server bundle license, it is not necessary to enter a license here.

Modifying Existing Domains and Servers

To modify existing Domains and Domain Management Servers, double-click the object in the SmartDomain Manager **General - Domain Contents** view. The pages and properties are identical to those for creating a new Domain ("Creating a New Domain Object" on page 76).

Working with Virtual Devices

When defining and managing virtual devices in Multi-Domain Security Management, you must use the SmartDashboard associated with a specific Domain Management Server. Otherwise, the configuration procedures are identical to those for a Security Gateway management model. Multi-Domain Security Management treats virtual devices much in the same manner as physical devices.

You can add as many Virtual Systems to Domain Management Servers as your license permits. Virtual Systems added to a Domain Management Server do not have to reside on the same VSX gateway or cluster. For more information regarding Domain Management Servers, refer to the *R75 Multi-Domain Security Management Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11683).

Adding Virtual System to a Domain Management Server

To add a new Virtual System to a Domain Management Server:

1. In the SmartDomain Manager, launch SmartDashboard from the appropriate Domain Management Server.
2. Create and configure the Virtual System ("Creating a New Virtual System" on page 42).
3. Define and install a security policy.

Adding Virtual Routers and Switches to a Domain Management Server

To add Virtual Routers and Switches to a Domain Management Server:

1. In the SmartDomain Manager, launch SmartDashboard from the appropriate Domain Management Server.
2. Create and configure Virtual Routers ("Creating a New Virtual Router" on page 55) and Virtual Switches ("Adding Virtual Switches" on page 51) as required.

Chapter 5

Introduction to VSX Clusters

This chapter presents a conceptual overview of VSX cluster deployments, with emphasis on clustering features and their application. This discussion assumes that the reader is familiar with network cluster applications and environments, particularly ClusterXL.

The Cluster Management chapter ("Managing VSX Clusters" on page 93) provides detailed configuration procedures, including instructions for enabling and using all VSX clustering features. For more about Check Point ClusterXL features and functionality, see the *R75.20 ClusterXL Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12265).

In This Chapter

VSX Clustering Overview	82
Planning a Cluster Deployment	83
VSX High Availability	85
Virtual System Load Sharing (VSLS)	86
Bridge Mode	90
Using Virtual Switches in a Cluster	92

VSX Clustering Overview

VSX clusters provide redundancy and load sharing features for Virtual Systems and other virtual devices. A VSX cluster consists of two or more identical, interconnected VSX gateways that ensure continuous data synchronization.

VSX high availability ensures continuous operation by means of transparent gateway or Virtual System failover. **Virtual System Load Sharing (VSLS)** enhances system performance by distributing active Virtual Systems amongst cluster members.

The advantages of using clusters in a VSX environment include:

- Transparent failover in case of gateway or Virtual System failure
- State synchronization ensures zero downtime for mission-critical environments
- Load sharing maintains system throughput during peak demand
- Enhanced scalability for future traffic growth

Physical Clusters

VSX clustering is based on Check Point ClusterXL concepts. This section reviews these concepts, and then demonstrates how these principles apply to VSX virtualization.

In typical Security Gateway deployment, a cluster consists of two or more identical, interconnected physical Security Gateways that provide redundancy and/or load sharing. This cluster behaves as a single Security Gateway and is assigned its own IP address, which is known as its **cluster IP** or **virtual IP**. This cluster IP address is distinct from the physical IP addresses of its cluster members, which are hidden from the networks connected to the cluster.

Traffic from external networks or the Internet directed to the internal networks arrives at the external cluster IP address. Depending on the clustering mode (high availability or load sharing), a designated cluster member receives the traffic and performs the required inspection. Following inspection, traffic is either sent to its destination on the internal network, or dropped.

Internal networks send traffic destined for the Internet or external networks, to the cluster IP address. This traffic is processed by the designated cluster member, inspected, and forwarded to its external destination.

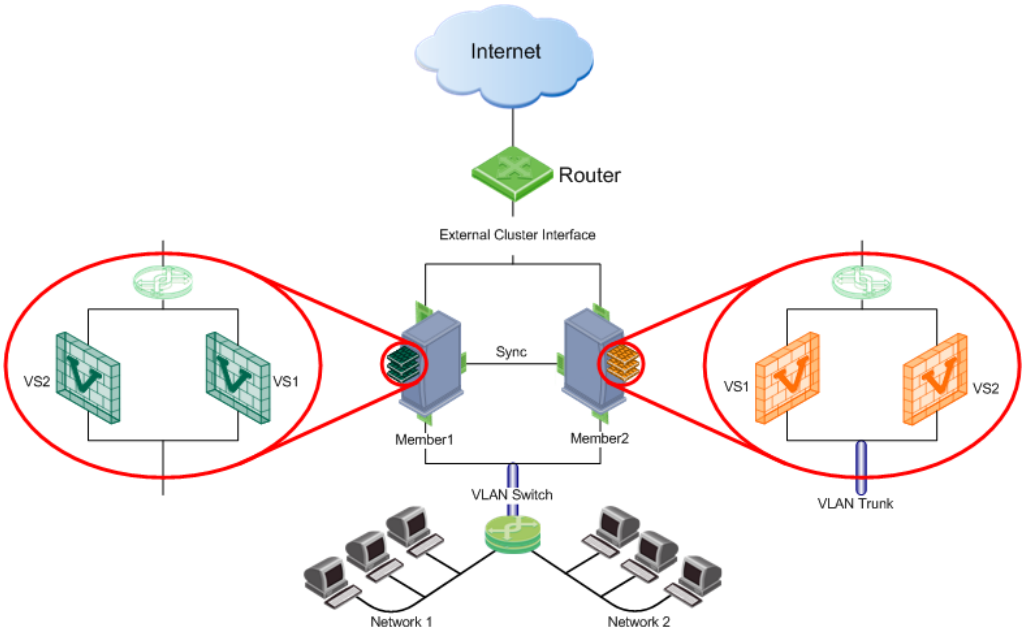
Each member interface has a unique, physical IP addresses. These IP addresses which are invisible to physical networks, are used for internal communication between members and the management server for such tasks as downloading policies, sending logs and checking the status of individual cluster members.

VSX Clusters

VSX clusters, like their physical counterparts, connect two or more synchronized gateways in such a way that if one fails, another immediately takes its place. VSX clusters are defined at two levels:

VSX ensures that Virtual Systems, Virtual Routers, Virtual Switches and their interfaces are provisioned and configured identically on each cluster member. The figure below shows that each cluster member contains identical instances of each virtual device. These identical instances are referred to as **peers**.

Figure 5-20 Typical Cluster Setup



VSX provides the management functionality to support network and security virtualization, including:

- **Assigning virtual IP addresses**: Each virtual device interface requires its own virtual IP address.
- **State synchronization**: Virtual device state tables are synchronized to peers on other cluster members.

Supported Cluster Environments

VSX supports the following proprietary clustering environments.

- Check Point ClusterXL
- Crossbeam Systems

The procedures presented in this Administration Guide focus on Check Point Cluster XL environments. If you are using one of the other supported environments, please refer to their documentation for assistance in implementing VSX cluster deployments.

Planning a Cluster Deployment

As with physical network deployments, advance planning is the key to successfully creating a working network. IP address allocation for a VSX deployment requires particular attention. This section takes you through the basics of IP address allocation for a VSX environment. Your VSX configuration choices affect the number of IP addresses required, both public and private.

VSX Cluster Architecture

VSX IP address allocation is similar to physical networks. Both real and virtual IP addresses are required for network connectivity (internal and external), management, and state synchronization.

VSX simplifies the IP address management task by automatically assigning IP addresses to Warp Links between virtual devices. For example, Warp Links between a Virtual Router and its associated Virtual Systems are created automatically and assigned IP addresses without user intervention.

A VSX cluster network contains the following components:

- Synchronization Network
- Internal Communications Network
- Virtual IP addresses

Synchronization Network

The synchronization network is a physical network that carries state synchronization data between cluster members. You configure the synchronization network during the initial VSX cluster definition and can make changes as necessary when adding or removing members.

State Synchronization can be used ClusterXL deployments as well as other OPSEC-certified VSX solutions. The synchronization network must be configured using unique IP addresses that are not used anywhere else in the enterprise network.

Internal Communication Network

The internal communication network is a virtual network that is required for Check Point ClusterXL environments in addition to the synchronization network. The internal communication network is invisible to external networks and enables cluster members to communicate and recognize the state of the environment.

VSX assigns an IP address to the internal communication network is assigned during the cluster creation process, eliminating the need to manually assign an IP addresses to each cluster member. The default IP address range consists of four class C networks:

```
IP address: 192.168.196.0

Net mask: 255.255.252.0
```

You can modify the default IP address using **Properties > Cluster members** page of the VSX cluster object, but only before creating Virtual Systems. Once Virtual Systems have been created, the IP range of the internal communication network cannot be modified.

Note - To avoid overlapping IP addresses, before creating any Virtual Devices, make sure the default IP address range of the Internal Communication network is not used anywhere else in the external network.

Virtual IP Addresses

Cluster (virtual) IP addresses are the only IP addresses visible to the external network. The assigned cluster IP addresses must correspond to the directly-connected subnet and server as a valid next hop address. These IP addresses are similar to virtual addresses configured across traditional cluster setups.

VSX High Availability

This section describes VSX high availability features. In a VSX environment, you can work with one of two high availability scenarios:

VSX Gateway High Availability: Each cluster member functions as a VSX gateway and is synchronized with the other members. If one member goes down, it immediately fails over to another member. Likewise, if an individual Virtual System, Virtual Router or Virtual Switch goes down, the entire member fails over to another member.

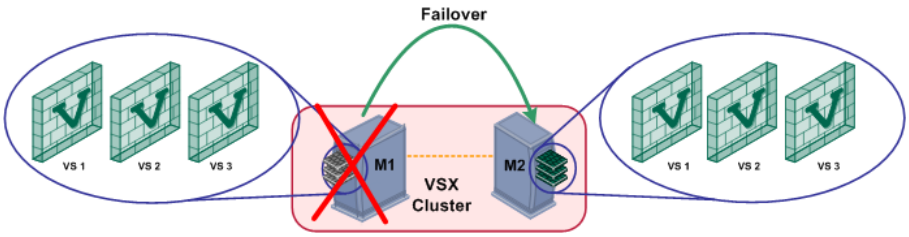
Per Virtual System High Availability: In the event that an individual Virtual System goes down, that Virtual System fails over to another member while all other Virtual Systems, together with other virtual devices, continue to function on the original member.

In either scenario, all members and virtual systems function in an active/active mode and are continuously synchronized.

VSX Gateway High Availability

VSX gateway high availability is the default cluster configuration. If neither Per Virtual System nor Virtual System Load Sharing (VSLS) is active, a cluster functions in the VSX Gateway high availability mode. All members of a cluster must be configured to use the same clustering mode.

Figure 5-21 VSX Gateway failover

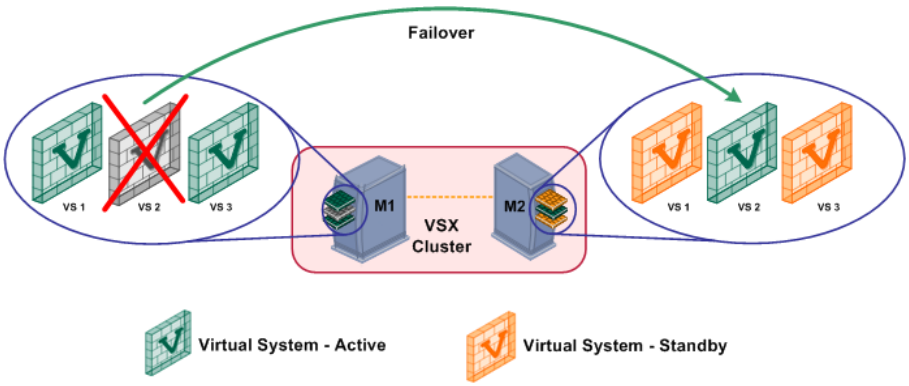


In the above example, member M1 experiences a failure the affects VS1 and all Virtual Systems immediately fail over to member M2.

Per Virtual System High Availability

With per Virtual System high availability, each Virtual System can monitor its own interfaces for failure, as illustrated in the figure below:

Figure 5-22 Virtual System failover



In this example, each member of the cluster contains three identical, synchronized Virtual Systems. The member designated as M1 process traffic. If VS2 goes down, on M1, it fails over to its peer in ton M2. VS1 and VS1 continue to function normally on M1.

For per Virtual System high availability to work properly, each Virtual System must connect directly to either a physical interface, or A VLAN.

Note - The following virtual devices are not supported when the Per Virtual System state is enabled:

- Virtual Routers
- Virtual Switches without physical or VLAN interfaces

Virtual System Load Sharing (VSLS)

VSX clusters can efficiently balance your network traffic load by distributing active virtual systems amongst cluster members. This capability is known as **Virtual System Load Sharing (VSLS)**, and provides the following benefits:

- **Capacity**: VSLS leverages the cluster machines to handle greater network volume by efficiently distributing the load.
- **Redundancy**: VSLS provides full redundancy by maintaining connectivity for all Virtual Systems even when individual members fail.
- **Scalability:** VSLS provides linear scalability for throughput and session rate.
- **Cost Effectiveness**: A VSLS cluster uses standard network switches to achieve cost effective load sharing.
- **Ease of Configuration**: Virtual Systems are automatically distributed among all the cluster members - no special configuration is required.
- **Priority Designation**: Mission-critical Virtual Systems can be separated from the other Virtual Systems, providing advantages in terms of bandwidth and resources.
- **System Scalability**: Every cluster member added to the cluster increases the overall system capacity and redundancy.

Note - The following virtual devices are not supported when the Per Virtual System state is enabled:

- Virtual Routers
- Virtual Switches without physical or VLAN interfaces

Requirements

- VSLS requires Check Point ClusterXL.
- VSLS requires that all Virtual Systems in all cluster members have direct connectivity with each other. Connectivity must be accomplished using switches or VLAN connections. This is required for detecting and assigning Virtual System states.
- VSLS does not support Virtual Routers.

Conceptual Overview

This section presents a detailed conceptual overview of ClusterXL Virtual System Load Sharing.

Introduction

Virtual System Load Sharing (VSLS) for VSX differs from physical cluster load sharing, in that it is not connection-based. Rather, its modus operandi is to distribute active Virtual Systems to different cluster members, and then direct traffic for particular Virtual System to the cluster member containing the active Virtual System. This is useful in balanced configurations, where the desire is to simply spread the load equally, and in mission-critical deployments, where reserving bandwidth for a particular Virtual System is a priority.

VSLS allows the administrator to either manually place specific Virtual Systems on specific cluster members, or allow the system to determine the dispersal configuration automatically. Refer to ("Configuring Virtual System Load Sharing" on page 113).

Note - You cannot configure a VSX ClusterXL in the Load Sharing mode if the cluster contains Virtual Systems in bridge mode or Virtual Routers.

Virtual System Priority

Virtual System priority refers to a preference regarding which member hosts a Virtual System's active, standby, and backup states. This preference is expressed as an integer value, as shown in the following table.

Priority	Definition
0	Highest priority, indicating the cluster designated to host the Virtual System's **active** state.
1	Second highest priority, indicating the member designated to host the Virtual System's **standby** state.
> 1	Lower priorities, indicating the members designated to host a Virtual System's **backup** state. The cluster member assigned priority 2 will be the first to switch the Virtual System to the Standby state in the event of a failure of either the Active or Standby Virtual System. A cluster member assigned priority 3 would be the next in line to come online in the event of another failure.

You can change the priority designation ("Distributing Virtual Systems Amongst Members" on page 115) using the `vsx_util vsls` command.

Virtual System Weight

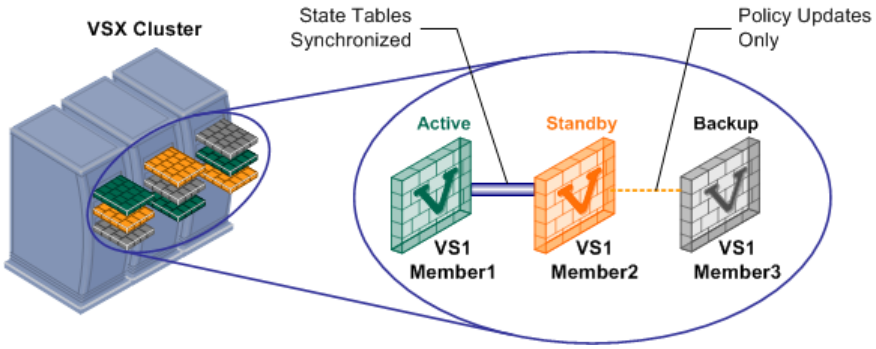
Since all Virtual Systems are not equal in terms of traffic and load, VSLS allows you to assign "weights" to individual Virtual Systems. The weight of a Virtual System affects the dispersal pattern of other Virtual Systems across cluster members. Assigning a heavier weight to a Virtual System gives it a larger share of a particular member's resources, and accordingly, disperses the other Virtual Systems to other cluster members.

By default, all virtual systems are assigned an equal weight factor of 10. You can change the weight factor ("Distributing Virtual Systems Amongst Members" on page 115) using the `vsx_util vsls` command.

Virtual System States

VSLS adds a backup state to the existing active and standby states. The backup state contains the latest configuration settings for each Virtual System, but does not receive state table synchronization. The relationship between Virtual System states is illustrated in the below figure.

Figure 5-23 State synchronization



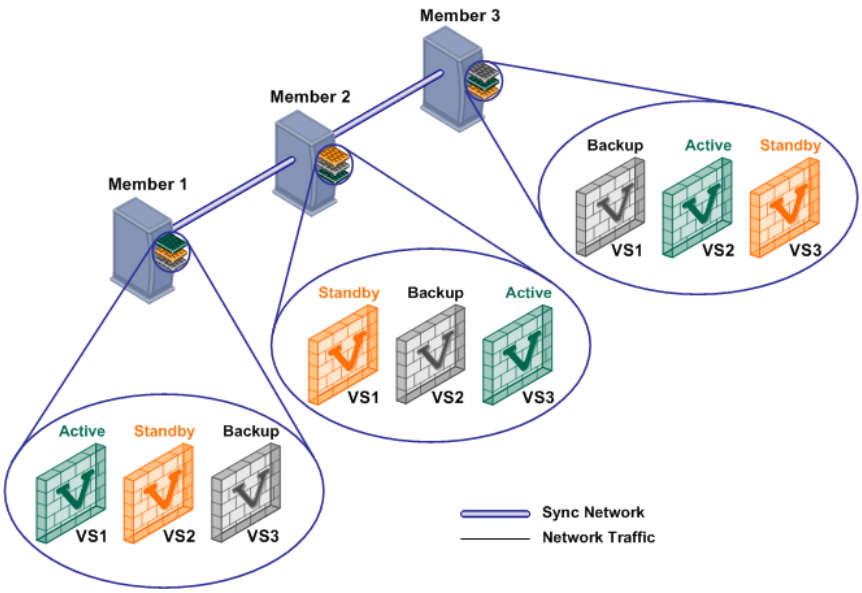
Each Virtual System peer in a VSLS cluster is replicated on all cluster members, and each copy exists in a different state. The active and standby states are synchronized so that the standby peer can immediately become active in the event of a failure of the active Virtual System or member. When this happens, the backup peer becomes the standby, and immediately synchronizes with the new active Virtual System.

VSLS reduces the load on the synchronization network by not synchronizing the backup Virtual System state tables with the active Virtual System until a failover occurs.

Normalized VSLS Deployment Scenario

The figure below illustrates a typical deployment scenario with three cluster members three, each containing three Virtual Systems. In this configuration, a an equalized load sharing deployment could have one active Virtual System on each cluster member.

Figure 5-24 Normalized VSLS deployment



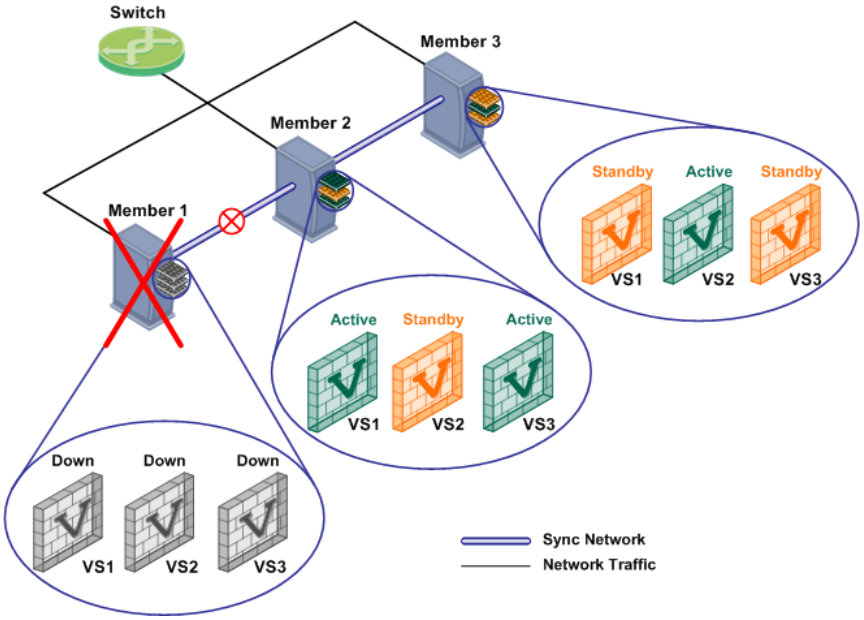
In the above figure, three Virtual Systems have been created and a different cluster member hosts the Active state of each. This distribution of Virtual Systems spreads the load among the clustered machines.

Once a Virtual System has been created, the system automatically creates Standby and Backup states and distributes them among the other cluster members.

Member Failure Scenario

In the event that a member fails or experiences a connectivity problem, VSLS detects the problem and routes traffic for the affected Virtual Systems to their respective standby Virtual Systems. Standby Virtual

Systems, which are fully synchronized with their active peers, change immediately to the active state and preserve active connections. At the same time, the backup Virtual Systems switch to standby, and synchronize fully with the newly active Virtual Systems.



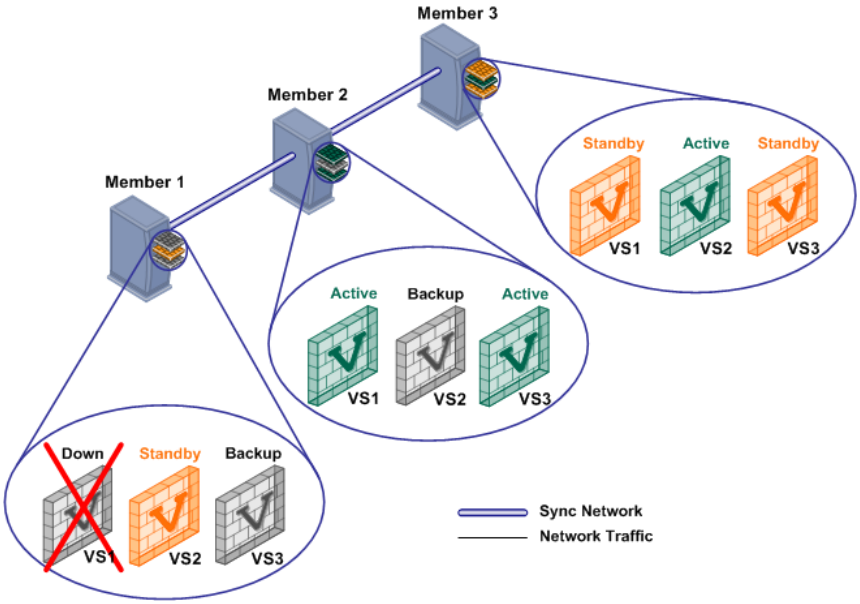
In this scenario, Member 1 fails and its active and standby Virtual Systems fail over to Members 2 and 3. The active Virtual System (VS1) moves to Member 2 and directs all VS 1 traffic itself. Its backup peer on Member 3 synchronizes with the new active Virtual System and becomes the standby.

VS2 on Member 2 becomes the standby and synchronizes with the active peer on Member 3. For VS3, the active and standby peers remain the same.

Virtual System Failure Scenario

The below figure illustrates failure scenario where an active Virtual System fails on one member, but the standby and backup Virtual Systems remain up. In this case, the active Virtual System fails over to its standby peer (in this case on Member 2) and its backup (on member 3) becomes the standby, synchronizing with the new active member.

Figure 5-25 Virtual System failover



All other virtual systems continue to function normally and no failover occurs.

Failure Recovery

When the failed cluster member or Virtual System comes back online, the system returns to its original load sharing configuration.

Bridge Mode

By implementing native layer-2 bridging instead of IP routing, you can add Virtual Systems without adversely affecting the existing IP structure.

When in the Bridge mode, Virtual System interfaces do not require IP addresses. You can optionally assign an IP address to the Virtual System itself (not the interfaces) to enable layer-3 monitoring, which provides network fault detection functionality.

VSX supports the following Bridge mode models:

- **STP Bridge Mode**: Provides redundancy while preventing undesirable loops between redundant switches.
- **Active/Standby Bridge Mode**: Provides path redundancy and loop prevention, while offering seamless support for Virtual System Load Sharing and overcoming many of the limitations of STP.

Spanning Tree Protocol (STP) Bridge Mode

The Spanning Tree Protocol is an industry standard technology designed to prevent loops in high-speed switched networks. To use the STP Bridge mode, you must have STP deployed and properly configured on you network. VSX supports the following STP layer-2 protocols:

- 802.1q
- 802.1D
- 802.1s
- 802.1w
- PVST+

Deploying and configuring STP on your network hardware is beyond the scope of this document. Please refer to your hardware documentation for assistance.

Active/Standby Bridge Mode

The Active/Standby Bridge mode enhances both *High Availability* and *Virtual System Load Sharing* in VSX clustered environments. By eliminating many disadvantages associated with the Spanning Tree Protocol (STP) bridge mode, Active/Standby Bridge mode provides significant improvements in High Availability deployments, while supporting Virtual System Load Sharing (VSLS).

Active/Standby Bridge mode offers the following advantages:

- Instantaneous failover
- Enhanced administrator control over bridge failover.
- VSLS support
- VLAN translation

The principal limitation of the Active/Standby bridge mode is that it breaks the STP tree structure.

Note - When configuring a Virtual System in the Active/Standby Bridge Mode, you should remove Virtual System VLANs from the STP database in the switches. This action prevents delays due to trunk interface failback.

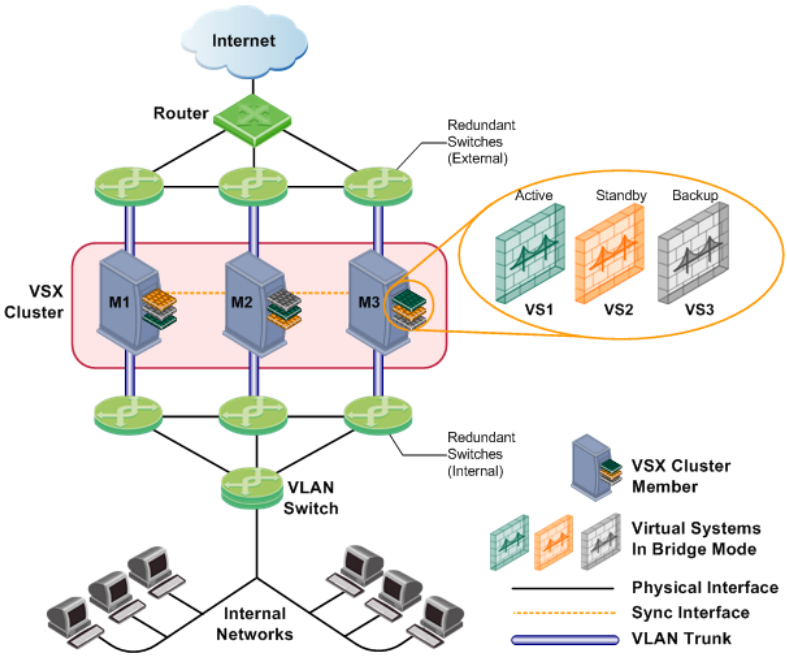
Deployment Scenarios

This section presents illustrative Active/Standby Bridge mode deployments, which cannot function using a standard STP Bridge mode configuration.

VLAN Shared Interface Deployment

In this deployment, each individual member connects to pair of redundant switches via a VLAN trunk. All Virtual Systems in a given member share the same VLAN trunk. The following figure illustrates example of such a deployment with active, standby and backup members.

Figure 5-26 Active/Standby bridge mode - shared interfaces



When using the Active/Standby Bridge mode in the High Availability mode, ClusterXL directs traffic to members according to administrator-defined priorities and status. In VSLS deployments, the system distributes the traffic load amongst members according to your VSLS configuration.

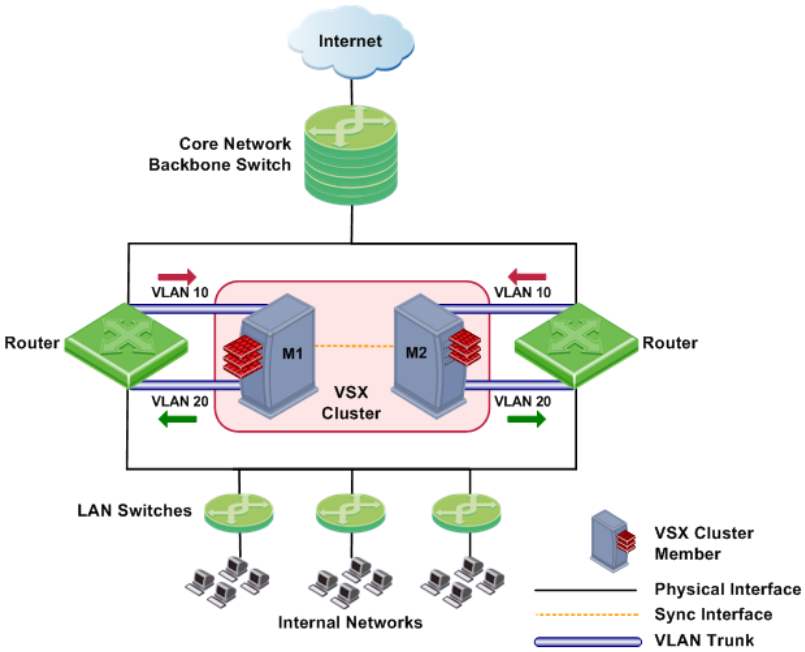
Three Layer Hierarchical Model

A three layer hierarchical model is used in very large, high-traffic network environments. It contains a mixture of components as described below:

1. A **core network**, comprised of high-speed backbone switches directs traffic to and from the Internet and other external networks.
2. A **distribution layer**, comprised of routers, provides connectivity between the core and the access layer.
3. An **access layer**, comprised of redundant LAN switches, forwards traffic to and from internal networks.

VSX, using the Active/Standby Bridge mode, is incorporated into the distribution layer, enforcing the security policy. This is illustrated in the following figure:

Figure 5-27 Active/Standby bridge mode - core network



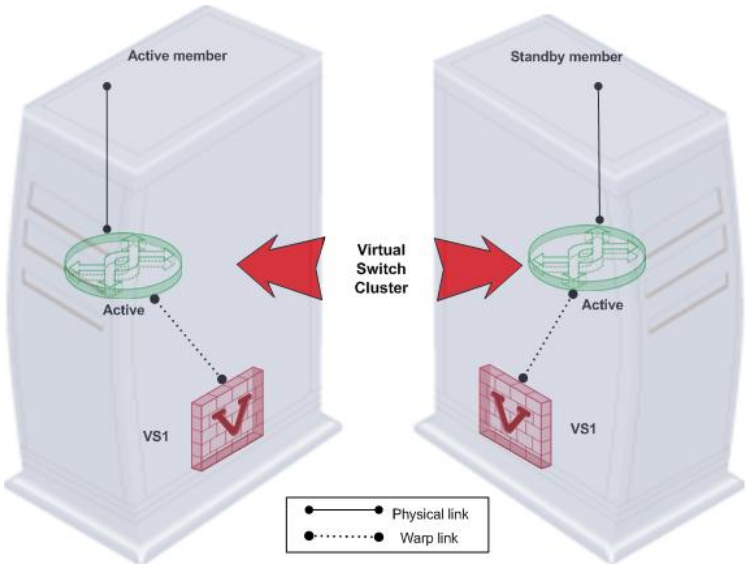
The routers direct external, "dirty" traffic to the appropriate Virtual System via a segregated VLAN. Filtered, "clean" traffic exits the Virtual System via a separate segregated VLAN back to the routers and on to internal destinations.

Using Virtual Switches in a Cluster

In a VSX cluster, Virtual Switches are also clustered for redundancy. Virtual Switches in the cluster are defined as *active/active.*

By means of the ClusterXL Control Protocol (CCP), the physical interface connected to the Virtual Switch is monitored. In the event of a failover, all Virtual Systems on *standby* become *active*, and send gratuitous ARPs from the warp interface between the Virtual System and the Virtual Switch.

Figure 5-28 Virtual Switches in a cluster



In the above figure, a simplified VSX cluster contains two members, one *active*, the other *standby*. The Virtual Switches within each cluster are *active/active*. When the physical interface connected to either Virtual Switch fails to respond, a failover occurs.

Chapter 6

Managing VSX Clusters

This chapter presents the procedures for configuring VSX in various cluster deployment scenarios. In addition to the basic scenarios, conceptual material and illustrative examples are presented for several advanced features, including the Bridge mode and dynamic routing.

In This Chapter

Configuration Overview	93
Creating a New Cluster	93
Modifying a Cluster Definition	98
Working with Cluster Members	107
Changing the Cluster Type	110
Configuring VSX High Availability	112
Configuring Virtual System Load Sharing	113
Configuring Virtual Systems in Bridge Mode	119
Advanced Clustering Configuration	122

Configuration Overview

The majority of the basic cluster configuration process is performed using SmartDashboard, both in Security Management and Multi-Domain Security Management models. However, you will need to use the command line interface to add additional members, remove members and upgrade existing members to VSX clusters. Many advanced cluster management, including load sharing definitions, require the command line.

Creating a New Cluster

This section describes how to create a new VSX cluster using the **VSX Cluster Wizard**. The wizard guides you through the following steps to configure a VSX cluster.

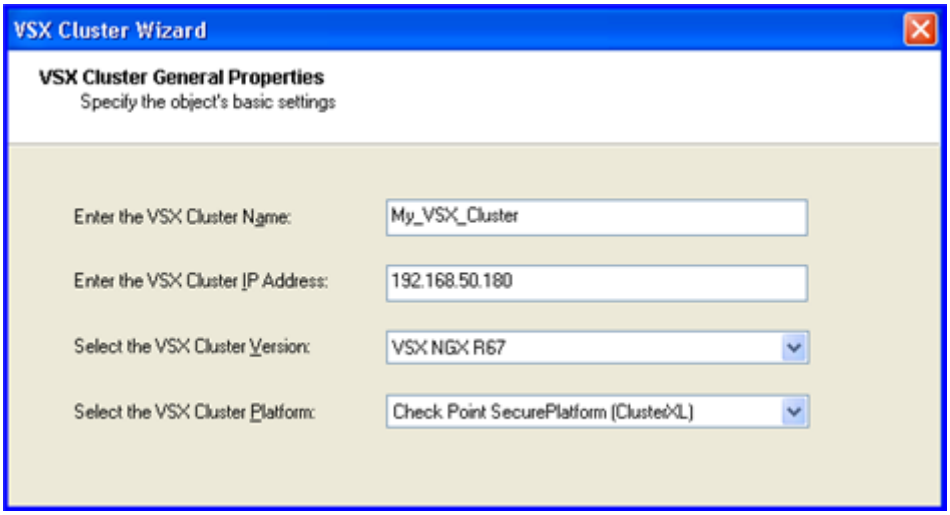
After completing the VSX Cluster Wizard, you can modify most cluster and member properties directly from SmartDashboard.

To create a new cluster:

1. Open SmartDashboard. If you are using Multi-Domain Security Management, open SmartDashboard from the Domain Management Server in which you are creating the cluster.
2. In the **Network Objects** tab in the **Objects Tree**, right-click **Check Point** and select **New Check Point > VPN-1 Power VSX >Cluster** . The VSX Cluster Wizard opens displaying the **General Properties** page.

Defining Cluster General Properties

The **Cluster General Properties** page contains basic identification properties for VSX gateways.



This window contains the following properties:

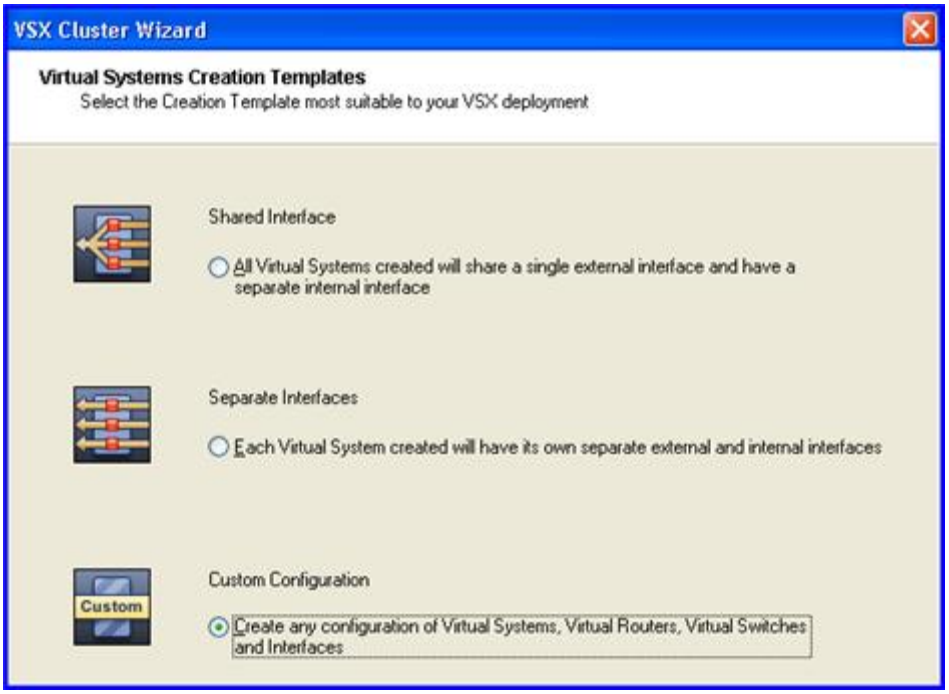
- **VSX Cluster Name**: Unique, alphanumeric for the cluster. The name cannot contain spaces or special characters except the underscore.
- **VSX Cluster IP Address**: Management interface IP address.
- **VSX Cluster Version**: VSX version to use for this cluster.
- **VSX Cluster Platform**: Platform type hosting the cluster members.
  - To create a HA cluster, select **Check Point SecurePlatform (ClusterXL)** from the list
  - To create a Load Sharing (VSLS) cluster, **Check Point ClusterXL Virtual System Load Sharing** select from the list.

Note - All cluster members must use the type of platform, with the same specifications and configuration.

Selecting Creation Templates

The **Virtual Systems Creation Templates** allows you to select a Virtual System Creation Template that automatically applies predefined, default topology and routing definitions to Virtual Systems when they are first created. This feature ensures consistency among Virtual Systems and speeds up the provisioning process.

You always have the option of overriding the default creation template when creating or modifying a Virtual System

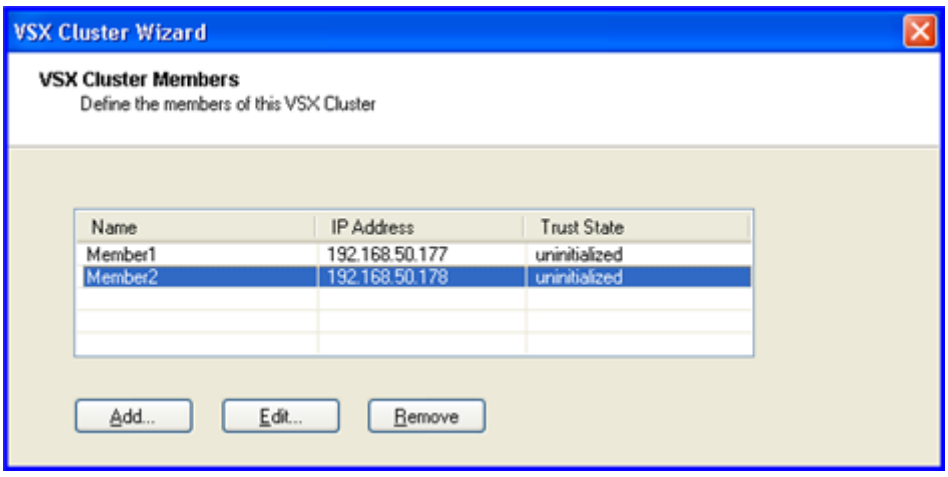

VSX Cluster Wizard
Virtual Systems Creation Templates
Select the Creation Template most suitable to your VSX deployment
Shared Interface
All Virtual Systems created will share a single external interface and have a separate internal interface
Separate Interfaces
Each Virtual System created will have its own separate external and internal interfaces
Custom
Custom Configuration
Create any configuration of Virtual Systems, Virtual Routers, Virtual Switches and Interfaces

The available creation templates are as follows:

- **Shared Interface**: All virtual systems share a single external interface, but maintain separate internal interfaces.
- **Separate Interfaces**: All virtual systems use their own separate internal and external interfaces. This template creates a Dedicated Management Interface (DMI) by default.
- **Custom Configuration**: You manually create a custom configuration without any template.

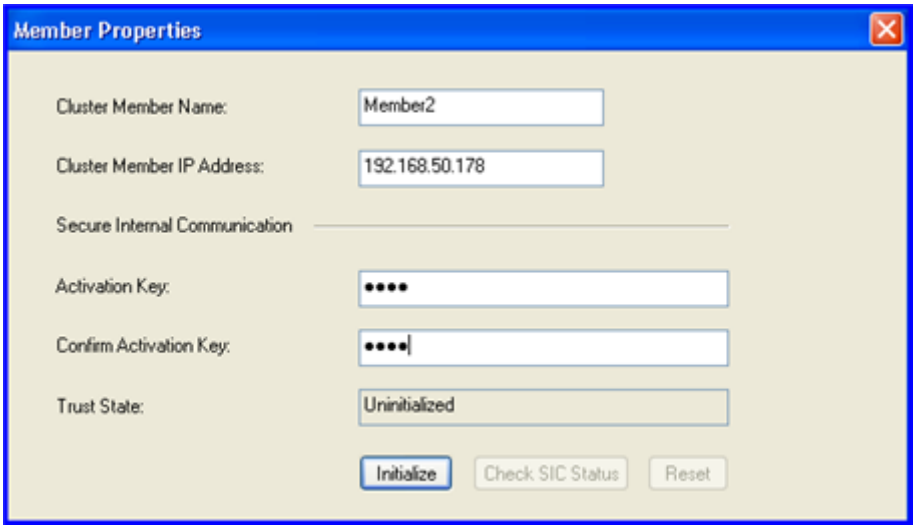
Adding Members

The VSX Cluster Members window defines the members of the new cluster. You must define at least two cluster here, and you can define as many as eight members. You can always add new member at a later time.



To add a new member:

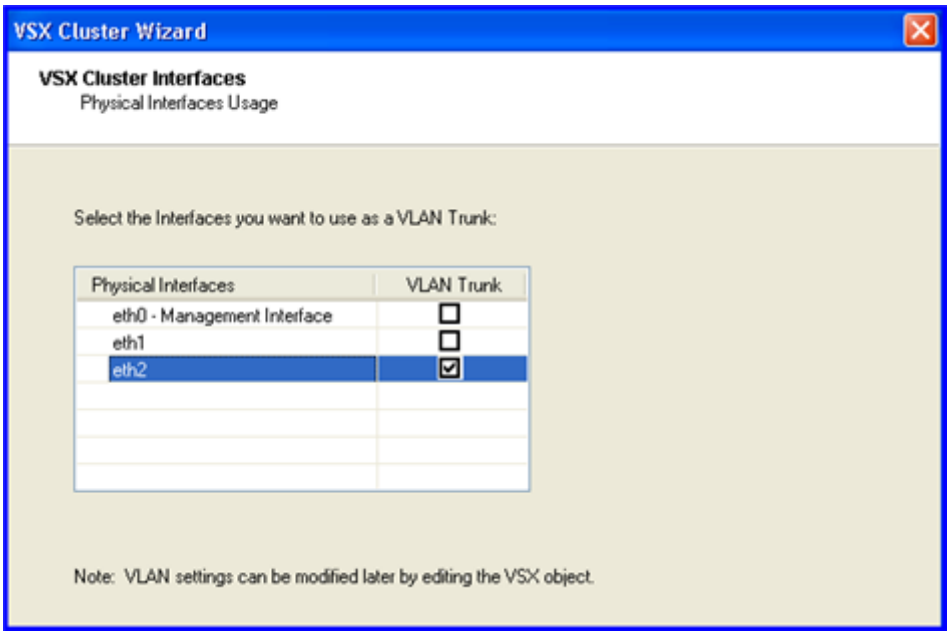
1. In the VSX Cluster Members window, click **Add**. The **Member Properties** window opens.



2. Enter the a unique member name and its IP address in the appropriate fields.
3. Enter and confirm the activation key to initialize SIC trust between the member and the management server.

Defining Cluster Interfaces

The **VSX Cluster Interfaces** window allows you define physical interfaces as VLAN trunks. The list displayed contains all interfaces currently defined on the gateway machine or cluster.

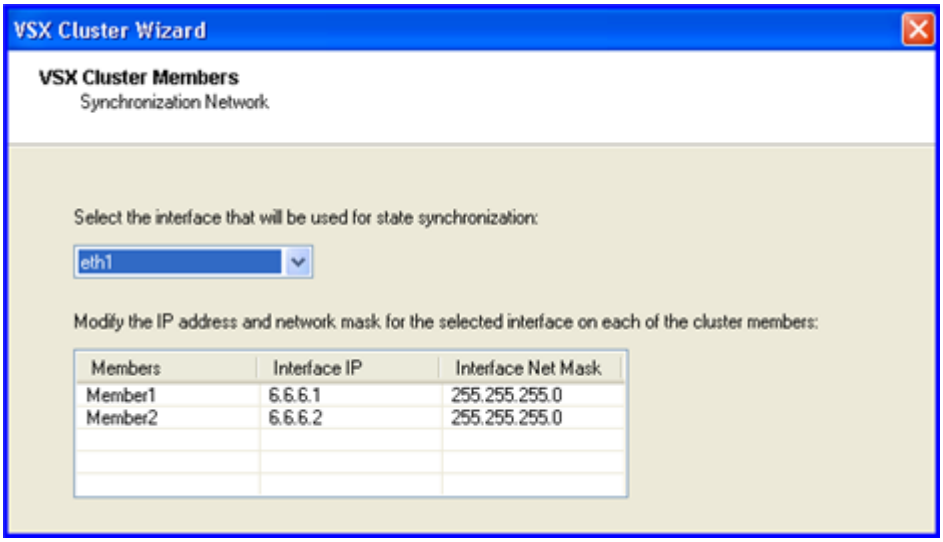

VSX Cluster Wizard
VSX Cluster Interfaces
Physical Interfaces Usage
Select the Interfaces you want to use as a VLAN Trunk:
Physical Interfaces
VLAN Trunk
eth0 - Management Interface
eth1
eth2
Note: VLAN settings can be modified later by editing the VSX object.

Select an interface to define it as a VLAN trunk. Clear an interface to remove the VLAN trunk assignment.

Important - Do not define the management interface as a VLAN trunk.

Configuring Cluster Members

If you selected the custom configuration option, the **VSX Cluster Members** window appears. In this window, you define the synchronization IP address for each member. .



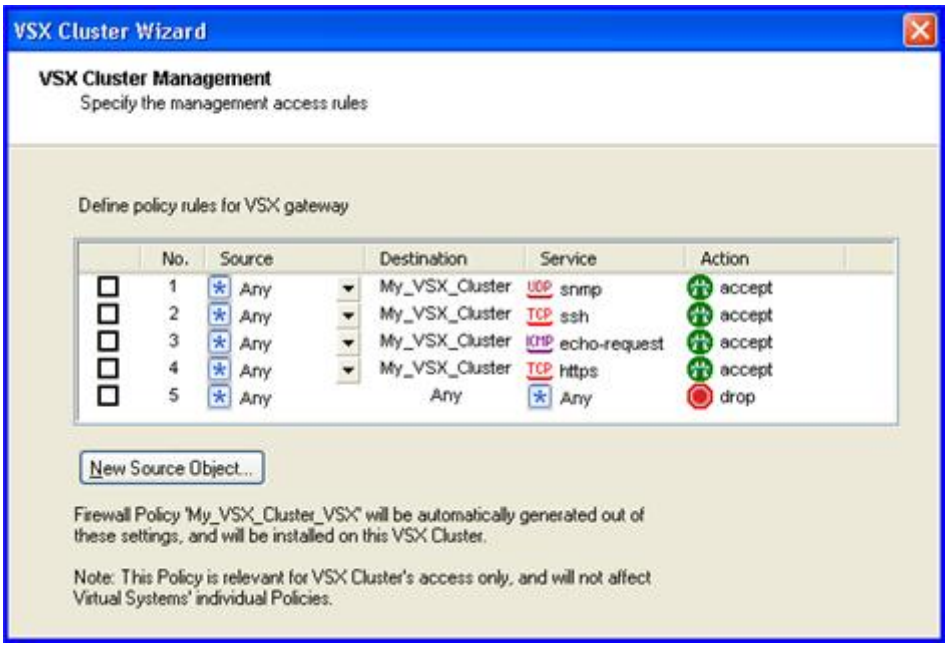
To configure the cluster members:

1. Select the synchronization interface from the list.
2. Enter the synchronization interface IP address and net mask for each member.

Cluster Management

The **VSX Gateway Management** page allows you to define several security policy rules that protect the cluster itself. This policy is installed automatically on the new VSX cluster.

Note - This policy applies **only** to traffic destined for the cluster. Traffic destined for Virtual Systems, other Virtual Devices, external networks, and internal networks is not affected by this policy.


VSX Cluster Wizard
VSX Cluster Management
Specify the management access rules
Define policy rules for VSX gateway
No.
Source
Destination
Service
Action
1 Any My_VSX_Cluster UDP snmp accept
2 Any My_VSX_Cluster TCP ssh accept
3 Any My_VSX_Cluster ICMP echo-request accept
4 Any My_VSX_Cluster TCP https accept
5 Any Any Any drop
New Source Object...
Firewall Policy 'My_VSX_Cluster_VSX' will be automatically generated out of these settings, and will be installed on this VSX Cluster.
Note: This Policy is relevant for VSX Cluster's access only, and will not affect Virtual Systems' individual Policies.

The security policy consists of predefined rules covering the following services:

- **UDP**: snmp requests
- **TCP**: ssh traffic
- **ICMP**: echo-request (ping)

- **TCP**: https (secure http) traffic

Configuring the Cluster Security Policy

1. **Allow**: Enable a rule to allow traffic for those services for which you wish to allow traffic. Clear a rule to block traffic. By default, all services are blocked.

   For example, you may wish to allow UDP echo-request traffic in order to be able to ping cluster members from the management server.
2. **Source**: Click the arrow and select a **Source Object** from the list. The default value is ***Any**. Click **New Source Object** to define a new source. Refer to the online help and the *R75 Security Management Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11667) for further details.

Completing the Wizard

1. Click **Next** to continue and then click **Finish** to complete the VSX Cluster wizard. Please note that this may take several minutes to complete. A message appears indicating successful or unsuccessful completion of the process.

   If the process ends unsuccessfully, click **View Report** to view the error messages. Refer to the troubleshooting steps ("VSX Diagnostics and Troubleshooting" on page 179) for more information
2. After the wizard finishes, make certain that the **Use State Synchronization** option is enabled in the **ClusterXL** branch of the **VSX Cluster Properties** window.

Modifying a Cluster Definition

Once you create a cluster using the wizard, you can modify the topology and other parameters using the **VSX Cluster Properties** window. This window also allows you to configure many advanced features not available with the wizard.

To work with a VSX cluster definition, double-click on the cluster object in the SmartDashboard Object Tree. The **VSX Cluster Properties** window opens, showing the **General Properties** page.

Most cluster objects and properties can be defined using the SmartDashboard GUI. Several definitions, however, require CLI commands, while others may be performed using either method.

A brief explanation for each of the definition pages follows. More detailed explanations for features that are not specific to VSX (NAT, IPS, VPN, etc.) are available in the online help or in the appropriate product Administration Guide.

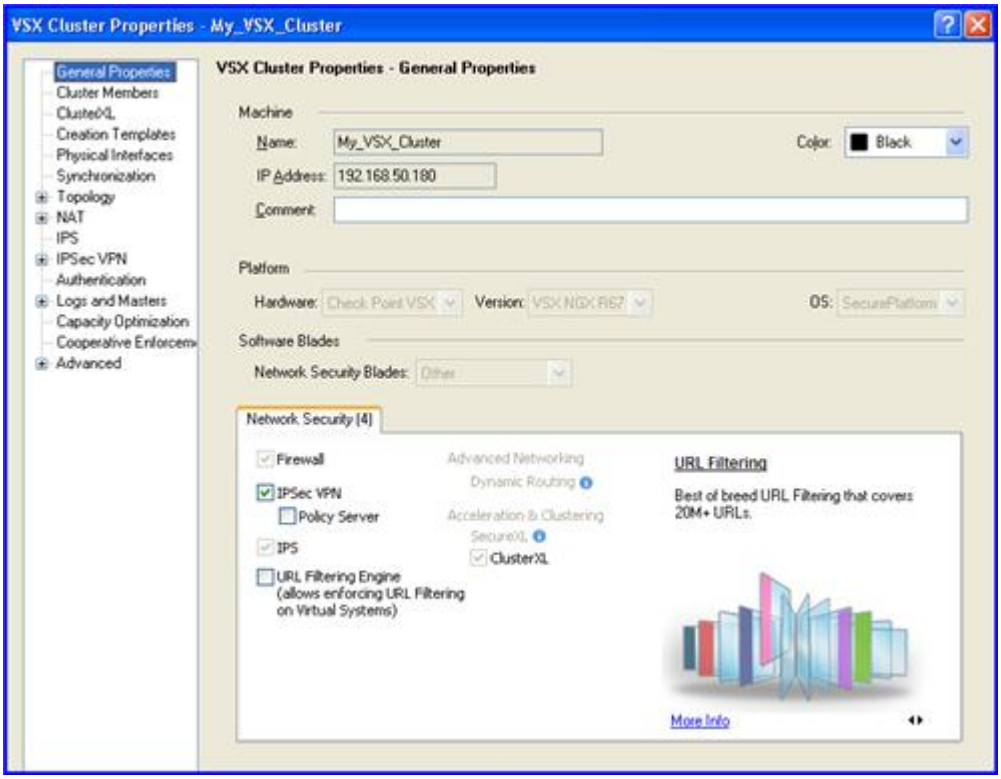
Modifying Cluster Properties

Once you create a cluster using the Wizard, you can modify its topology and other parameters using the **VSX Cluster Properties** window. This window also allows you to configure advanced features that are not available with the wizard.

To work with a VSX cluster definition, double-click on the VSX gateway object in the SmartDashboard Object Tree. The **VSX Cluster Properties** window opens, showing the **General Properties** page.

General Properties

Use the **General Properties** page to view general properties and to activate Check Point products for use with this cluster and its members.

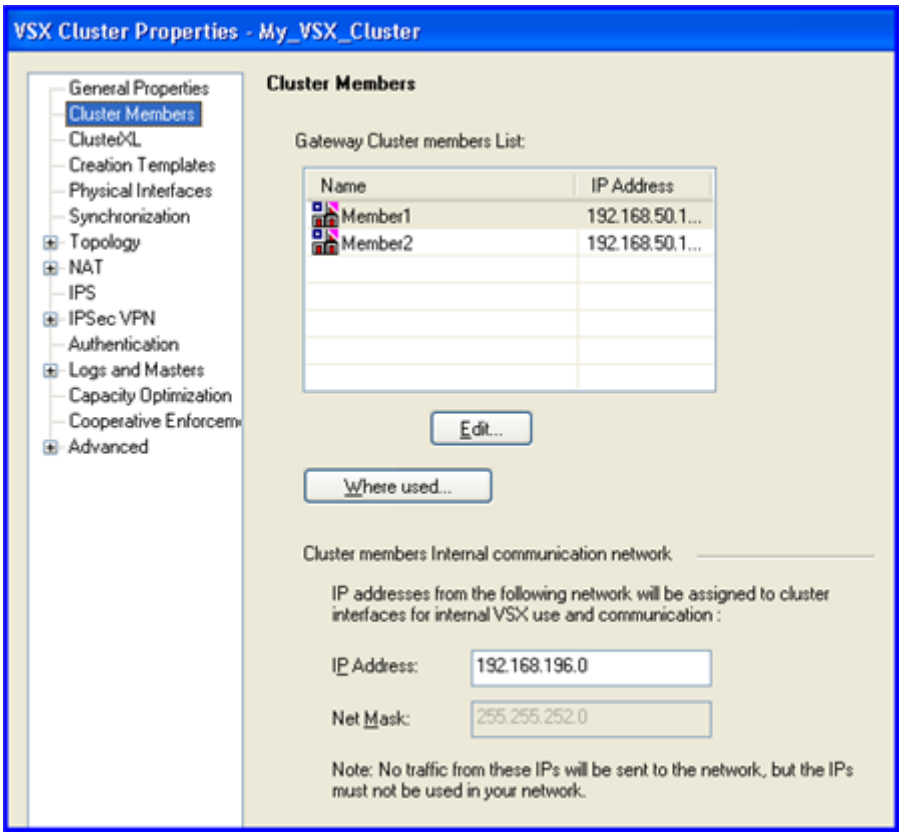


You can modify the following properties:

- **Comment**: Free text comment that appears in the Object List and elsewhere
- **Color**: Color of the object icon as it appears in the Object Tree
- **Check Point Products**: Select Check Point products active on this cluster and its members

Cluster Members

The **Cluster Members** page enables you to view and/or modify several properties for individual cluster members, including IP addresses for members and the internal communication network. You can also view where cluster and member objects in the object database are used.



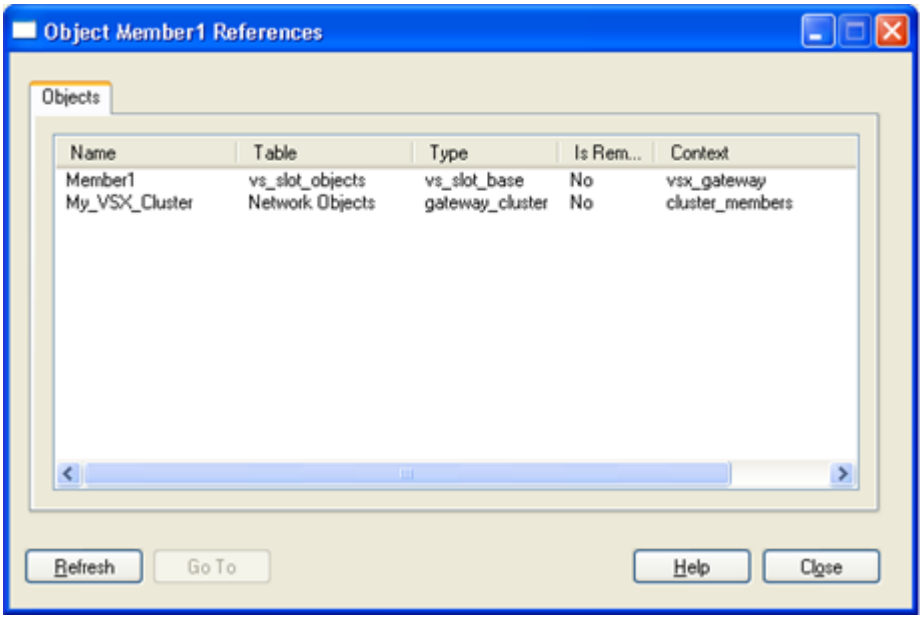
Gateway Cluster Member List

This tab shows the currently defined cluster members. Double-click on a member or select a member and click **Edit** to open its **Cluster Member Properties** window. You can modify the following member properties on various tabs as follows:

- **General Tab**:
  - **Comment**: Free text comment that appears in the Object List and elsewhere
  - **Color**: Color of the object icon as it appears in the Object Tree
  - **Secure Internal Communication**: Check and reset SIC trust
  - **Topology Tab**: Displays the member IP address and net mask for each interface. Double click on an interface to displays its properties.
  - **NAT Tab**: Allows you to define NAT ("Virtual System - NAT" on page 49) rules for cluster members connected to a Virtual Router.
  - **VPN Tab**: Contains a variety of configuration properties for site-to-site VPN deployments. This window is only available if the Check Point VPN product is enabled on the **General Properties** page.
  - Please refer to the online help and the *R75 VPN Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11675) for further details regarding VPN concepts and configuration.

Where Used

Click **Where used** to display information relating to the selected member in the objects database. The following data appears in the window:



- **Name**: Cluster name.
- **Table**: Name of the table in the database under which the selected object is listed.
- **Is removable**: Specifies whether or not you are allowed to remove the selected object. If the object is not removable and nevertheless you choose to remove it, it will impact the database or rule base.
- **Refresh**: Click to update the window display if you make changes.
- **Context**: Where the object is used.

Internal IP Address and Net Mask

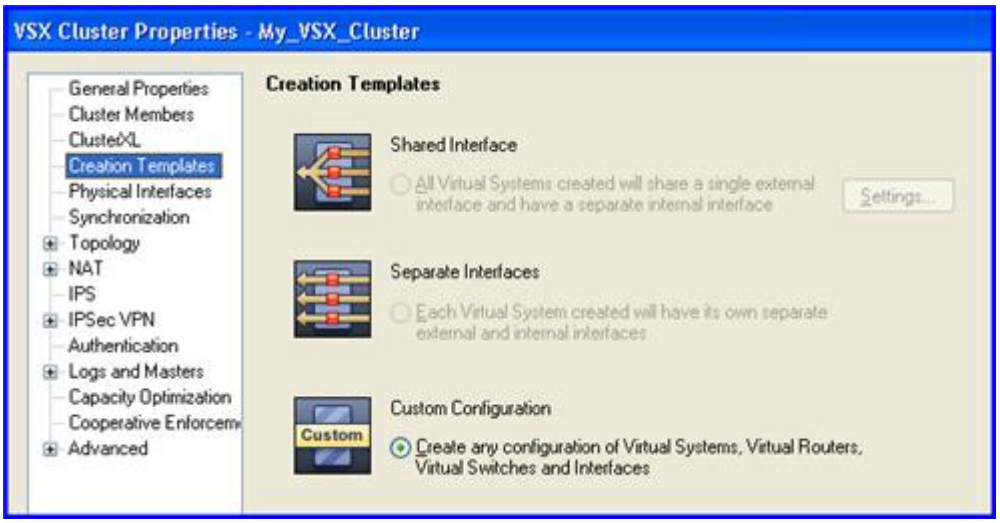
VSX creates an internal communication network and automatically assigns it an IP address and net mask from a predefined pool. You can change this IP address here if you have not yet defined a Virtual System. Although traffic from this address is never sent to any networks, you must ensure that this IP address is unique and not in use anywhere on your defined network.

ClusterXL

You can enable or disable state synchronization in the **ClusterXL** window and choose options to track changes in the state of cluster members on this page. All other properties are ClusterXL configuration properties are disabled. You can modify the Cluster XL configuration using the `vsx_util` command.

Creation Templates

The **Creation Templates** page displays the creation template used to create Virtual Systems. You can change from the current creation template to the **Custom Configuration** template and change the shared physical interface if the Shared Interface template is active.



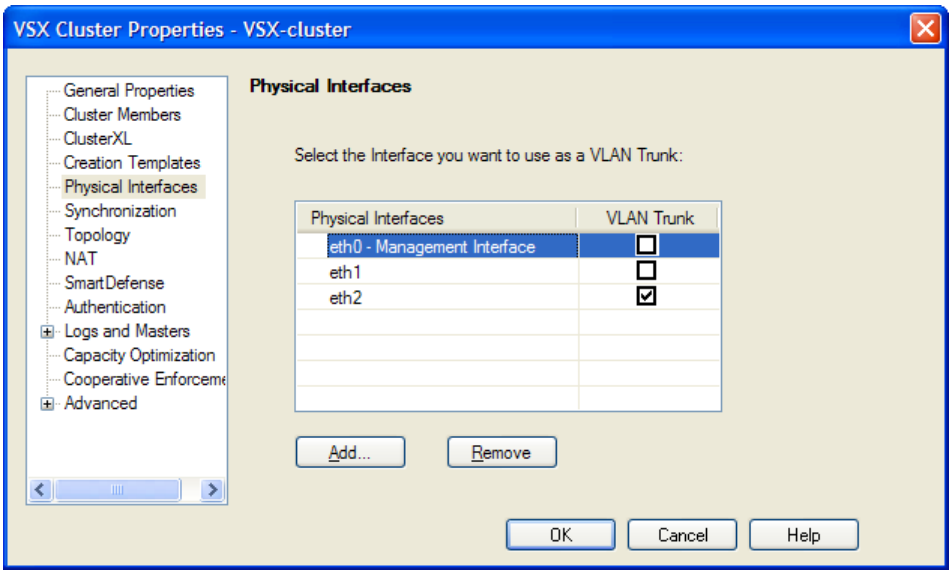
- Select the **Custom Configuration** option to change from the Shared Interfaces or Separate Interfaces templates.

  You cannot change back from the **Custom Configuration** template once you have completed the definition and saved it to the configuration to cluster.

- To change the shared interface, click **Settings** and select an interface from the list that appears.

Physical Interfaces

The **Physical Interfaces** page allows you to add or delete a physical interface on the VSX gateway, and to define interfaces to be used as VLAN trunks.



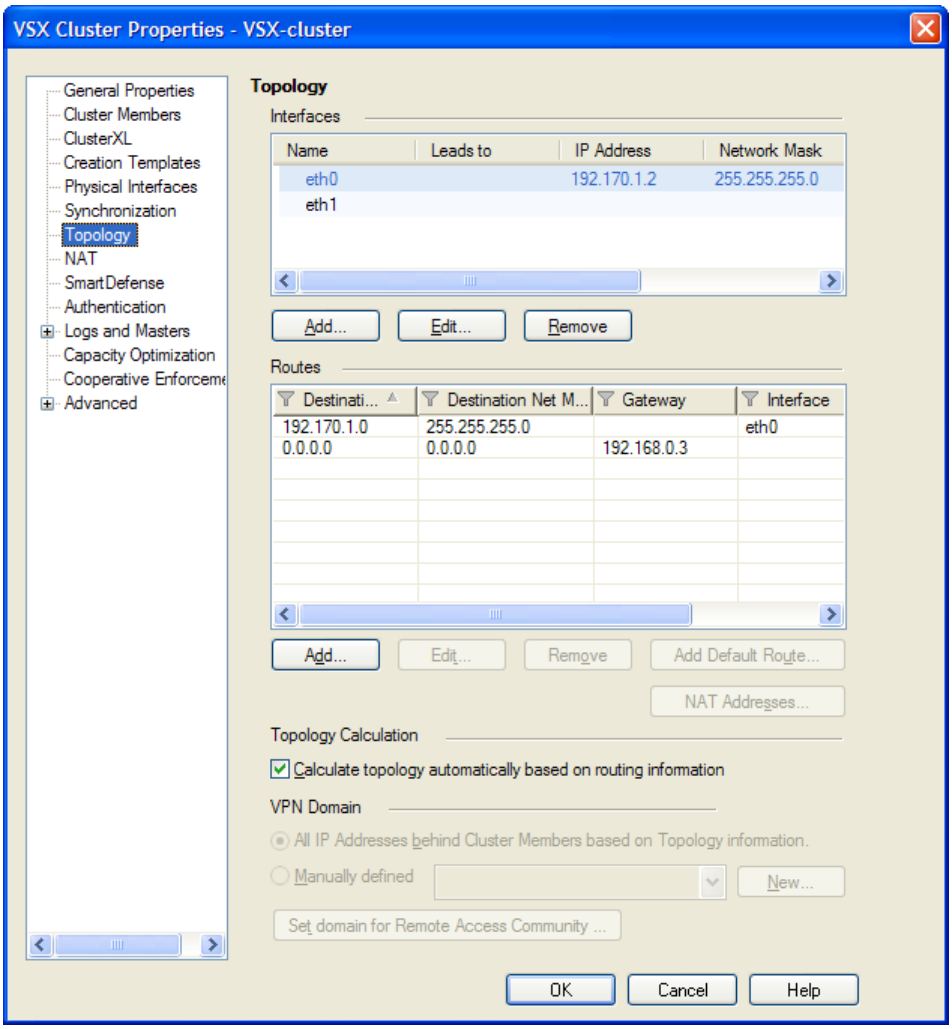
- To add a new physical interface, click **Add** and enter the interface name in the appropriate field.
- To define an interface as a VLAN trunk, select the desired interface and enable the VLAN Trunk option. To disable a VLAN trunk, clear the option.

Synchronization

The **Synchronization** window displays the state synchronization network. There are no configurable properties.

Topology

The **Topology** page contains interface and routing definitions.



Interfaces

The Interfaces section defines interfaces and links to devices. You can add new interfaces as well as delete and modify existing interfaces.

To add an interface:

1. Click **Add**.
2. In In the **Interface Properties** window, select an interface from the list and define the appropriate properties ("Modifying an Interface Definition" on page 63).

The **Cluster Members** tab in the **Interface Properties** window contains cluster member IP address and net mask definitions each interface.

To Modify an interface:

1. Double-click an interface.
2. In the **Interface Properties** window, enter an IP address and/or net mask value.
3. Modify other parameters as required.

Routes

The Routes section defines routes between Virtual Systems, external network devices, network addresses, and other virtual devices. Some routes are defined automatically based on the interface table. You can add new interfaces as well as delete and modify existing interfaces.

To add a default route to the routing table:

1. Click **Add Default Routes** and either enter the default route IP address or select the default router.

2. In the **Route Configuration** window, modify the IP address, net mask and next hop parameters as necessary.
3. Enable or clear the **Propagate route to adjacent Virtual Devices** option as necessary.

Click **Help** for details regarding the various properties and options.

Calculating topology automatically based on routing information

Enable this option to allow VSX to automatically calculate the network topology based on interface and routing definitions (enabled by default). VSX creates a automatic links, or connectivity cloud objects linked to existing internal or external networks.

- This option is not available in the Bridge mode.
- When employing dynamic routing, it is recommended to disable this option.

VPN Domain

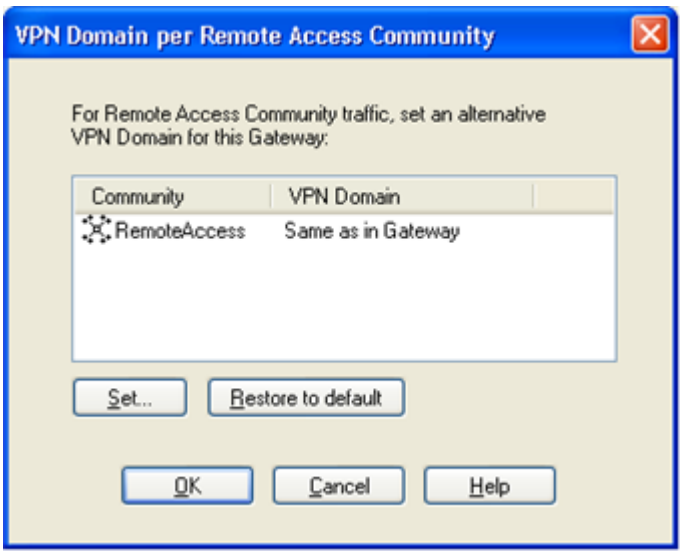
The VPN Domain defines the set of hosts located behind a given Virtual System that communicate via a VPN tunnel with peer Virtual Systems.

When including a virtual device as part of a VPN connection, you must specify a VPN Domain. The domain definition specifies those Virtual System interfaces that are included in the VPN. You can define a VPN Domain in one of two ways by enabling the appropriate option:

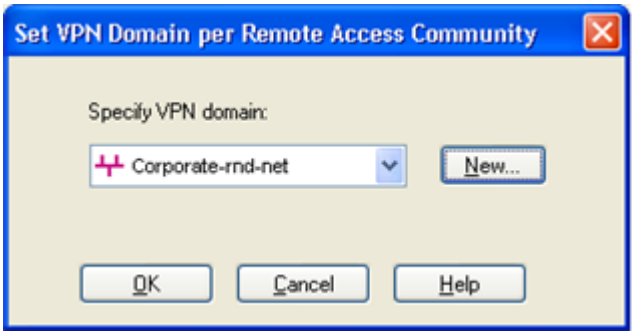
- **All IP Addresses behind gateway based on topology information**: Includes all hosts not located behind an external gateway cluster interface.
- **Manually Defined**: Includes all hosts in the selected network or group.
- **Remote Access Communities**: You can optionally specify an alternative VPN domain for Remote Access Community traffic.

To specify the VPN domain:

1. Click **Set domain for Remote Access Community**.
2. In the **VPN Domain per Remote Access Community** window, double-click a Remote Access Community.



In the **Set VPN Domain** window, select a VPN domain from the list or click **New** to define a new domain. Click **OK** in both windows to continue.



NAT

The **Advanced** page allows you to configure NAT for Virtual Systems connected to a Virtual Router.

To enable and configure NAT:

1. Enable the **Add Automatic Address Translation** option.
2. Select a translation method from the list.
   - **Hide NAT**: Hide NAT only allows connections originating from the internal network. Internal hosts can access internal destinations, the Internet and other external networks. External sources cannot initiate a connection to internal network addresses.
   - **Static NAT**: Static NAT translates each private address to a corresponding public address.
3. If you select **Hide NAT**, select one of the following options:
   - **Hide behind Gateway** hides the real address behind the VSX gateway external interface address. This is equivalent to hiding behind the address 0.0.0.0, or
   - **Hide behind IP Address** hides the real address behind a virtual IP address, which is a routable, public IP address that does not belongs to any real machine.
4. If you selected **Static NAT**, enter the static IP address in the designated field.
5. Select the desired VSX cluster from the **Install on Gateway** list.

IPS

VSX uses the default protection profile. There are no configurable properties on this page.

VPN

The **VPN** page contains a variety of configuration properties for clusters in site-to-site VPN deployments. This window is only available if the Check Point VPN product is enabled on the **General Properties** page.

Please refer to the online help and the *R75 VPN Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11675) for further details regarding VPN concepts and configuration.

Remote Access

The Remote Access page contains properties that govern establishing VPN connections with Remote Access clients. This window is only available if the Check Point VPN product is enabled on the **General Properties** page.

Please refer to the online help and the *R75 VPN Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11675) for further details regarding VPN with Remote Access clients.

Authentication

The **Authentication** page allows you to enable several different authentication options ("Working with Authentication" on page 63) for a VSX gateway.

Logs and Masters

The **Logs and Masters** page allows you define logging ("Tracking Activity with SmartView Monitor" on page 69)options for a VSX gateway.

Capacity Optimization

The Capacity Optimization page allows you to maximize cluster and VPN throughput by limiting the number of concurrent connections, the number of concurrent IKE negotiations, and the number of concurrent VPN tunnels. To raise or lower the maximum, use the arrows in the appropriate field to set the desired value.

Cooperative Enforcement

Cooperative Enforcement works with Check Point Endpoint Security servers. This feature utilizes the Endpoint Security server compliance capability to verify connections arriving from various hosts across the internal network. The Cooperative Enforcement window contains several configuration properties for defining this feature. For more information, please refer to the online help and the *R75 IPS Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11663).

Advanced Pages

The **VSX Bridge Configuration** page allows you to specify the loop detection algorithm when working in the Bridge mode.

Enable the **Check Point ClusterXL** option to enable the **Active/Standby Bridge mode** loop detection algorithms contained in ClusterXL.

Enable the **Standard Layer-2 Loop Detection Protocols** to use standard loop detection protocols, such as STP or PVST+.

For more about SNMP, connection persistence and permissions to install policies, see the *R75.20 Firewall Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12267) and the *R75.20 IPS Administration Guide (*http://supportcontent.checkpoint.com/documentation_download?ID=12270*).*

Changing the Cluster Management IP and/or Subnet

You can change the cluster management IP address and/or subnet by executing the `vsx_util change_mgmt_ip` ("change_mgmt_ip" on page 198) and `vsx_util change_mgmt_subnet` ("change_mgmt_subnet" on page 201) commands.

Changing the Internal Communication Network IP

You can change the internal communication network IP address by using the *vsx_util change_private_net* ("change_mgmt_private_net" on page 198) command.

Working with Cluster Members

This section presents procedures for adding and deleting cluster members, as well as for upgrading existing cluster members to VSX.

Adding a New Member

Important - Verify that no other administrators are connected to the management server before proceeding. The `vsx_util` command cannot modify the management database if the database is locked because other administrators are connected.

To add a new member to an existing cluster:

1. Close SmartDashboard and backup the management database.
2. Enter the Expert mode.
3. From the command prompt on the management server, enter the expert mode and run the `vsx_util add_member` command. Perform the following steps prompted:
   a) Enter the Security Management Security Management Server or main Domain Management Server IP address.
   b) Enter the administrator name and password.
   c) Enter the name of the VSX cluster.
   d) Enter the name of the new member.
   e) Enter the member management interface IP address and net mask.
   f) Enter the Sync interface IP address and net mask.
   g) Enter any other interface IP addresses as may be required
4. Wait until the **add member operation finished successfully** message appears, indicating that the database has been successfully updated and saved.

   Note - In a Multi-Domain Security Management environment, this operation will skip any Domain Management Servers locked by an administrator. If this should occur, run the operation again for the relevant Domain Management Servers once they become available.
5. Open SmartDashboard and verify that an object representing the new member appears in the specified cluster. Modify its configuration as required. Close SmartDashboard.
6. From the management server command prompt, Enter the Expert mode and run the `vsx_util add_member_reconf` command. Enter the following information when prompted:
   a) Security Gateway or main Domain Management Server IP address
   b) Administrator name and password
   c) New member name
   d) SIC activation key for the new member
7. Wait until the **Reconfigure module operation completed successfully** summary notice appears.

   Note - In a Multi-Domain Security Management environment, the operation will skip any Domain Management Servers locked by an administrator. If this should occur, run the operation again for the relevant Domain Management Servers when they become available.

8. Reboot the new member.

If the cluster is running in the VSLS mode, run `vsx_util vsls` to redistribute Virtual Systems to the newly added member.

Deleting a Member

Important - Verify that no other administrators are connected to the management server before proceeding. The `vsx_util` command cannot modify the management database if the database is locked.

You perform this operation using the management server command line. It is strongly recommended that you back up the database prior to removing a member.

To remove a member from a cluster:

1. Detach the license from the member to be removed. You cannot remove a member if the license is attached.
2. Close SmartDashboard.
3. From the management server command line, run the `vsx_util remove_member` command. Perform the following tasks as prompted:
   a) Enter the Security Gateway or main Domain Management Server IP address.
   b) Enter the administrator name and password.
   c) Type '`y`' to confirm that you have detached the license from the member.
   d) Enter the cluster name.
   e) Enter the cluster member name.
   f) Type '`y`' to confirm that the member to be removed has been disconnected.
4. Wait until the `remove member operation finished successfully` message appears. The database is now updated and saved. In SmartDashboard, the object for the deleted member no longer appears in the specified cluster.

   Note - In a Multi-Domain Security Management environment, the operation will skip any Domain Management Servers locked by an administrator. If this should occur, run the operation again for the relevant Domain Management Servers when they become available.
5. Open SmartDashboard and verify that deleted member no longer appears in the specified cluster.

Upgrading Cluster Members

This section describes the procedures for upgrading cluster members that were initially installed using an earlier version of VSX. You perform the upgrade process using the `vsx_util upgrade` command. Afterwards, you use the `vsx_util reconfigure` command to apply settings stored in the management database to the newly upgraded member.

Preliminary Steps

Perform the following steps before attempting to upgrade a cluster member:

1. Verify that your management server is version is currently running VSX. For detailed information on upgrading your management server, please refer to the *R75 Installation and Upgrade Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11648).
2. If you intend to use a different machine for the upgraded gateway, verify the following:
   - The new machine meets the minimum hardware requirements as defined in the *Release Notes*.
   - The new machine contains at least the same number of interfaces as the old machine, and that each interface has the same name and IP address as before.
   - If you intend to upgrade a member on the same machine, backup the existing member to a spare machine. For detailed information on backing up a VSX member, see the backup and restore

section of the *R75 SecurePlatform Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11666).

Upgrading a Member to the Current Version

Important - Verify that no other administrators are connected to the management server before proceeding. The `vsx_util` command cannot modify the management database if the database is locked.

Performing the following steps to upgrade the cluster and its members:

1. Close SmartDashboard.
2. Enter the Expert mode.
3. Execute the `vsx_util upgrade` command from the management server command line.

   Enter the following information when prompted:

   a) Security Gateway or main Domain Management Server IP address

   b) Administrator name and password

   c) Cluster name
4. When prompted, select the version to which you wish to upgrade.
5. Wait until the `Finished upgrading/database saved successfully` message appears, indicating that the database has been updated and saved.
6. Open SmartDashboard and verify that an object representing the new member now appears in the specified cluster.

   Note - In a Multi-Domain Security Management environment, the operation will skip any Domain Management Servers locked by an administrator. If this should occur, run the operation again for the relevant Domain Management Servers when they become available.
7. Perform a fresh installation of VSX on each upgraded member.
8. Perform the initial configuration steps on each member as described in the *R75 Installation and Upgrade Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=11648), including:

   a) Define the IP address, net mask and default gateway.

   b) Install a valid license.

   c) Set the SIC activation key.

   d) Configure the cluster properties as required. These property settings must be the same as defined for the other cluster members.
9. Run the `vsx_util reconfigure` command from the management server command line. Enter the following information when prompted:

   a) Management server or main Domain Management Server IP address

   b) Administrator name and password

   c) SIC activation key for the upgraded member

   This action installs the existing security policy and configuration on the newly upgraded members.
10. Wait until the `Finished upgrading/database saved successfully` message appears.

    Note - In a Multi-Domain Security Management environment, the operation will skip any Domain Management Servers locked by an administrator. If this should occur, run the operation again for the relevant Domain Management Servers when they become available.
11. Reboot each member.

Notes to the Upgrade Process

- You only need to run the `vsx_util upgrade` command once for each VSX cluster. You must, however, run the `vsx_util reconfigure` command for each cluster member.

  For example, for a deployment with two clusters, each cluster having three members, run `vsx_util upgrade` twice, once for each cluster *object*, and the `vsx_util reconfigure` six times, once for each cluster *member*.

- To ensure stability of the VSX deployment, run the `vsx_util upgrade` and `vsx_util reconfigure` consecutively. For example, run the `vsx_util upgrade` command for the cluster object first and then run the `vsx_util reconfigure` for the first member**.** Run `vsx_util reconfigure` for the second and subsequent members immediately thereafter·
- Verify that each upgraded member is fully operational before upgrading the remaining members.
- You cannot install policies until the upgrade process has completed successfully for all members.

Changing the Cluster Type

This section presents procedures for converting cluster members from one cluster type (High Availability or VSLS) to the other. Changing the cluster mode involves the use of the `vsx_util convert_cluster` command.

Converting from VSLS to High Availability

The process of converting a cluster from VSLS to High Availability consists the following procedures:

1. Redistributing all active Virtual Systems to one member
2. Disabling VSLS options
3. Converting the cluster to High Availability

Redistributing Active Virtual Systems to One Member

To redistribute all active Virtual Systems to one member:

1. Close SmartDashboard.
2. Enter the Expert mode.
3. Execute the `vsx_util vsls` command.
4. Enter the Security Management Server or Multi-Domain Security Management Domain Management Server IP address.
5. From the Load Sharing menu, enter "**3. Set all VSs active on one member**".
6. Enter the administrator user name and password.
7. Enter the VSX cluster name.
8. Enter the number corresponding to the member designated to host all active members.
9. Enter "**y**" to save and apply the configuration.
10. Exit the Load Sharing menu.

When the `convert_cluster` command finishes, there should be only one active member on which all Virtual Systems are in the **active** state, and one standby member on which all Virtual Devices are in the **standby** state. Any additional members should be in **standby** mode and their Virtual Devices in the **down** state.

Disabling VSLS Options

To convert existing cluster members to the VSX gateway high availability mode:

1. On each member, execute the `cpconfig` command and do the following:

   a) Disable the Per Virtual System State for each member.

   b) Disable ClusterXL for Bridge Active/Standby for each member.

2. Re-initialize the members using the `cpstop` and `cpstart` commands.

Converting the Cluster

To convert the cluster to HA:

1. Execute the `vsx_util convert_cluster` command .
2. Enter the Security Management Server or Multi-Domain Security Management Domain Management Server IP address.
3. From the Load Sharing menu, enter "**3. Set all VSs active on one member**".
4. Enter the administrator user name and password.
5. Enter the VSX cluster name.
6. Enter "**HA**"
7. Re-initialize all members using the `cpstop` and `cpstart` commands.

Converting from High Availability to VSLS

To convert an existing high availability cluster to VSLS load sharing, perform the following steps:

1. Close SmartDashboard.
2. On each member, execute the `cpconfig` command and do the following:
   a) Enable the Per Virtual System State for each member.
   b) Enable ClusterXL for Bridge Active/Standby for each member.
3. Re-initialize the members using the `cpstop` and `cpstart` commands.
4. On the management server, enter the Expert mode.
5. Execute the `vsx_util convert_cluster` command.
6. Enter the Security Management Server or Multi-Domain Security Management Domain Management Server IP address.
7. Enter the administrator user name and password.
8. Enter the VSX cluster name.
9. Enter "**LS**"
10. At the "*Proceed with conversion?*" prompt, enter "**y**".
11. Choose an option for distributing Virtual Systems amongst members:
    a) Distribute all Virtual Systems so that each member is equally loaded.
    b) Set all Virtual Systems as **Active** on the same member.

Note - You cannot convert a VSX cluster to the VSLS mode if it contains Virtual Systems in the STP Bridge mode or Virtual Routers.

Sample Command Output

The following screen printout shows an example of the output from the `vsx_util convert_cluster` command.

```
vsx_util convert_cluster

**********************************************
Note: the operation you are about to perform changes the
information in the management database. Back up the database
before continuing.
**********************************************
Enter Security Management Server/main Domain Management
Server IP address (Hit 'ENTER' for 'localhost'):
Enter Administrator Name:
Enter Administrator Password:
Enter VSX cluster object name:
Enter desired ClusterXL mode: HA-High Availability, LS-Load
Sharing (HA | LS): LS

All modules must be in the 'Per VS State' mode to conclude
this operation successfully. Use the command cpconfig on each
module to verify compliance before continuing with the
operation.

When converting a cluster, there are two options for
distributing the existing Virtual System(s) among cluster
members:
1. Distribute all Virtual Systems so that each cluster member
is equally loaded.
2. Set all Virtual Systems as Active on the same cluster
member.

After converting the cluster, the command vsx_util
redistribute_vsls may be used to modify Virtual System
distribution.
Enter distribution option (1-2) [1]: 1

Converting the cluster to ClusterXL Load Sharing mode...
The cluster was successfully converted to ClusterXL Load
Sharing mode
Installing new policy...
...
Policy installation finished successfully.
```

Configuring VSX High Availability

This section presents procedures for configuring VSX high availability features. In a VSX environment, you can work with one of two high availability modes:

VSX Gateway High Availability: Each cluster member functions as a VSX gateway and is synchronized with the other members. If a member goes down, it immediately fails over to another member. Likewise if an individual Virtual System, Virtual Router or Virtual Switch goes down, the entire member fails over to another member.

Per Virtual System High Availability: In the event that an individual Virtual System goes down, that Virtual System fails over to another member while all other Virtual Systems, together with other virtual devices, continue to function on the original member.

Enabling VSX Gateway High Availability

VSX gateway high availability is the default cluster configuration. If neither Per Virtual System nor Virtual System Load Sharing (VSLS) is active, a cluster functions in the VSX Gateway high availability mode. All members of a cluster must be configured to use the same clustering mode.

Configuring New Cluster Members

To configure members for VSX gateway high availability:

1. During the initial configuration phase (`configsys`), enter '`y`' in response to the "`Would you like to install a Check Point clustering product?`" question to enable VSX clustering.
2. Enter '`n`' when prompted to enable the Per Virtual System State.
3. Select the "ClusterXL" option.
4. Configure all other properties according to your requirements.

Enabling Per Virtual System High Availability

This section describes the procedures for enabling the Per Virtual System Failover in a high availability cluster.

Note - The following virtual devices are not supported when the Per Virtual System state is enabled:

- Virtual Routers
- Virtual Switches without physical or VLAN interfaces

Configuring New Cluster Members

To configure members for Per Virtual System high availability:

1. During the initial configuration phase, enter '`y`' in response to the "`Would you like to install a Check Point clustering product?`" question to enable VSX clustering.
2. Enter '`y`' when prompted to enable the Per Virtual System State.
3. Enter '`n`' when prompted to enable Virtual System Load Sharing.

Configure all other properties according to your requirements.

Configuring Virtual System Load Sharing

This section presents the various procedures for configuring VSLS deployments. You use the *vsx_util vsls* to perform various VSLS configurations tasks.

To start vsx_util vsls:

1. From the management server Expert mode, execute *vsx_util vsls*.
2. Enter the management server IP address.
3. Enter administrator user name and password.
4. Enter the VSX gateway name.
5. From the VSLS menu, choose the desired option.

Enabling VSLS

In order to use VSLS for VSX, you must first enable the Per Virtual System State mode on each cluster member. You can then create a Load Sharing cluster, either by creating a new cluster object, or by converting an existing High Availability cluster to Load Sharing mode. After completing this process, you can modify Virtual Systems as required.

Note - To use the latest Load Sharing features on existing VSLS clusters, you must upgrade all cluster members to the current version. You should also apply the hotfix to the management server.

Enabling the Per Virtual System State Mode

The **Per Virtual System State** mode enable active Virtual Systems to be placed on different cluster members, and for Virtual System-specific failover. This setting is mandatory for VSLS. On each cluster member, do the following:

Note - The following virtual devices are not supported when the Per Virtual System state is enabled:

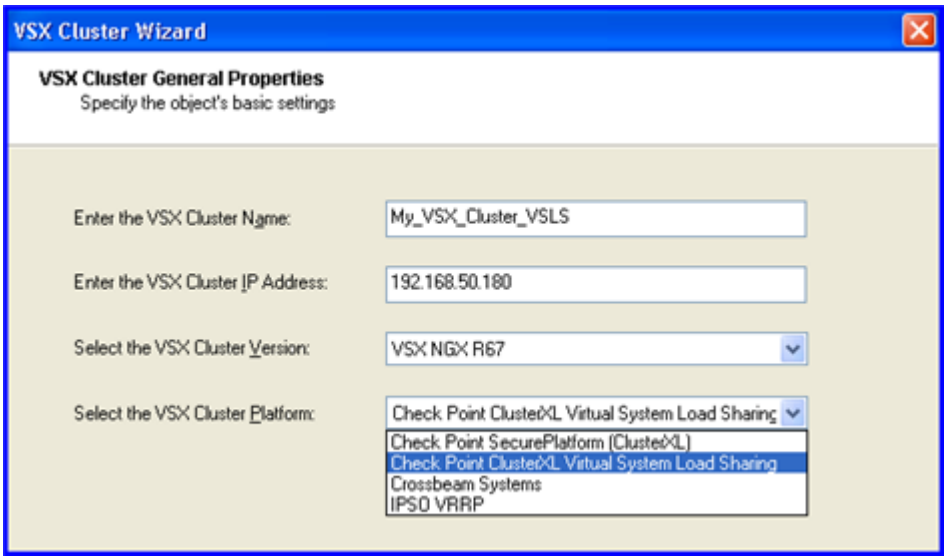
- Virtual Routers
- Virtual Switches that do not have physical or VLAN interfaces

1. Run `cpconfig`.
2. Select `Enable Check Point Per Virtual System State`.
3. Answer `y` to the question: `Would you like to enable Per Virtual System state?`
4. Reboot the machine.
5. Repeat this procedure for each member.

Creating a New VSLS Cluster

To create a new VSLS cluster:

1. From the Objects Tree in SmartDashboard, right click on **Check Point** and select **New Check Point > VPN-1 Power VSX > Cluster**. The **VSX Cluster Wizard** opens.
2. Create and configure the new cluster ("Creating a New Cluster" on page 93).
   a) On the **General Properties** page, select **Check Point ClusterXL Virtual System Load Sharing** from the **VSX Cluster Platform** list.


VSX Cluster Wizard
VSX Cluster General Properties
Specify the object's basic settings
Enter the VSX Cluster Name:
My_VSX_Cluster_VSLS
Enter the VSX Cluster IP Address:
192.168.50.180
Select the VSX Cluster Version:
VSX NGX R67
Select the VSX Cluster Platform:
Check Point ClusterXL Virtual System Load Sharing
Check Point SecurePlatform (ClusterXL)
Check Point ClusterXL Virtual System Load Sharing
Crossbeam Systems
IPSO VRRP

   b) On the **Creation Templates** page, select creation templates ("Selecting Creation Templates" on page 94).
   c) Continue the cluster definition process according to your requirements.

Using the vsx_util vsls Command

You use the *vsx_util vsls* command to perform various Virtual System Load Sharing configuration tasks, including:

1. Displaying the current VSLS configuration
2. Distributing Virtual Systems equally amongst cluster members
3. Set all Virtual Systems as active on one member
4. Manually define the priority and weight for individual Virtual Systems
5. Import VSLS configurations from comma separated value (CSV) text files

6. Export VSLS configurations to comma separated value (CSV) text files
7. Exporting and Import VSLS configurations from/to comma separated value (CSV) text files

To work with the vsx_util vsls command:

1. Run *vsx_util vsls f*rom the Expert mode on the management server
2. Select the desired choice from the **VSLS** menu

```
Enter Administrator Name: aa
Enter Administrator Password:
Enter VSX cluster object name: vsx

VS Load Sharing - Menu
______________________________
1. Display current VS Load sharing configuration
2. Distribute all Virtual Systems so that each cluster member is
equally loaded
3. Set all VSs active on one member
4. Manually set priority and weight
5. Import configuration from a file
6. Export configuration to a file
7. Exit

Enter redistribution option (1-7) [1]:
```

Distributing Virtual Systems Amongst Members

The primary advantage of VSLS is the ability to distribute active, standby and backup Virtual Systems amongst cluster members in order to maximize throughput and user response time. You can choose to distribute Virtual Systems according to one of the following options:

- Automatically distribute active Virtual Systems amongst cluster members so that all members are equally loaded based on assigned weights and existing or default priority definitions.
- Automatically place all active Virtual Systems on the same member
- Manually define priorities and weights for each Virtual System

Distributing Virtual Systems for Equal Member Loading

To distribute Virtual Systems for equal member loading:

1. From the VSLS menu, select "**2. Distribute all Virtual Systems so that each cluster member is equally loaded**".
2. At the "**Save & apply configuration ?**" prompt, enter "**y**" to continue.

The process update process may take several minutes or longer to complete, depending on the quantity of Virtual Systems and cluster members.

Placing All Active Systems on the Same Member

1. From the VSLS menu, select "**3. Set all VSs active on one member**".
2. When prompted, enter the number corresponding to the member designated as the primary member.
3. When prompted, enter the number corresponding to the member designated as the standby member. All other members will be designated as backup members.
4. At the "**Save & apply configuration ?**" prompt, enter "**y**" to continue.

The process update process may take several minutes or longer to complete, depending on the quantity of Virtual Systems and cluster members.

Manually Assigning Priorities and Weights for a Single Virtual System.

You can modify these settings in one of two ways:

- Automatically assign weights only to Virtual Systems. This method prompts you for a weight for each Virtual System and then automatically updates the settings.
- Manually assign both priorities and weights to individual Virtual Systems.

To Automatically assign weights to all Virtual Systems:

1. From the VSLS menu, select "**Manually set priority and weight**".
2. Enter "**a**" to automatically scroll through each Virtual System.
3. For each Virtual System, enter a weight value and press **Enter**.
   a) If you do not enter a weight value for a Virtual System, the currently assigned weight is retained.
   b) To stop entering weight values for additional Virtual systems, enter "**s**". Only those Virtual Systems that have been assigned a new weight value will be updated.
4. At the "**Save & apply configuration ?**" prompt, enter "**y**" to continue.

The process update process may take several minutes or longer to complete, depending on the quantity of Virtual Systems and cluster members.

To manually assign priorities and weights for individual Virtual Systems:

1. From the VSLS menu, select "**Manually set priority and weight**".
2. Enter "**m**" to manually update both priorities and weights for individual Virtual Systems.
3. At the "**Would you like to change the virtual system's priority list?**" prompt, enter '**y**' to change the member priority.
   a) Enter the number associated with the member to receive the highest priority.
   b) Enter the number associated with the member to receive the next highest priority.
   c) Continue until all members have been assigned a priority.
4. At the "**Would you like to change the virtual system's weight?**" prompt, enter "**y**" to assign a weight "**n**" to retain the existing weight value.
   a) At the prompt, enter an integer between 1 and 100, representing the new weight value.
5. At the "**Do you wish to configure another Virtual System?**" prompt, enter "**y**" to configure another Virtual System or "**n**" continue.
6. At the "**Save & apply configuration ?**" prompt, enter "**y**" to continue.

The process update process may take several minutes or longer to complete, depending on the quantity of Virtual Systems and cluster members.

Viewing VSLS Status

To view the current VSLS status and Virtual System distribution amongst members, select "**1. Display current VS Load sharing configuration**" from the VSLS menu. The output is similar to the below example"

```
----+---------+-----------+-----------+-----------+--------+
VSID| VS name | gw150     | gw151     | gw152     | Weight |
----+---------+-----------+-----------+-----------+--------+
  2 | vs1     | 0         | 1         | 2         | 10     |
  3 | vs2     | 2         | 0         | 1         | 10     |
  4 | vs3     | 1         | 2         | 0         | 10     |
  5 | vs5     | 0         | 2         | 1         | 10     |
  6 | vs4     | 1         | 0         | 2         | 10     |
----+---------+-----------+-----------+-----------+--------+
 Total weight | 20        | 20        | 10        | 50     |
----+---------+-----------+-----------+-----------+--------+

Legend:
0 - Highest priority
1 - Next priority
2 - Lowest priority
```

Virtual System Priority

Virtual System priority refers to a preference regarding which member hosts a Virtual System's active, standby, and backup states. This preference is expressed as an integer value, as shown in the following table.

Priority	Definition
0	Highest priority, indicating the member designated to host the Virtual System **active** state.
1	Second highest priority, indicating the member designated to host the Virtual System **standby** state.
> 1	Lower priorities, indicating members designated to host a Virtual System's **backup** state. The cluster member assigned priority 2 will be the first to switch the Virtual System to the Standby state in the event of a failure of either the Active or Standby Virtual System. A cluster member assigned priority 3 would be the next in line to come online in the event of another failure.

Virtual System Weight

Each Virtual System is assigned a weight factor, which indicates its traffic volume relative to the total traffic volume (the sum of all weight factors) on a given cluster member. VSX uses the weight factor to determine the most efficient distribution of Virtual Systems amongst cluster members. System resource allocation is not affected by the weight factor, nor does VSX take weight into consideration for any other purpose.

By default, all virtual systems are assigned an equal weight factor of 10.

Exporting and Importing VSLS Configurations

When working with large scale VSLS deployments consisting of many Virtual Systems, multiple cluster members, using the vsx_util command to perform configuration tasks can be quite time consuming. To allow administrators to efficiently configure such deployments, VSX supports uploading VSLS configuration files containing configuration information for all Virtual Systems directly to management servers and cluster members.

This capability offers the following advantages:

- Rapid VSLS configuration of large-scale deployments with many Virtual Systems and cluster members.
- Efficient migration and scalability for complex deployments
- External backup of VSLS configurations.

VSLS configuration files are comma separated value (CSV) files that are editable using a text editor or another applications, such as Microsoft Excel. You can use the configuration file to rapidly change the weight and cluster member priority for each Virtual Systems in the list.

Note - You cannot use the VSLS configuration file to add or remove cluster members. You must use the appropriate vsx_util commands to accomplish this.

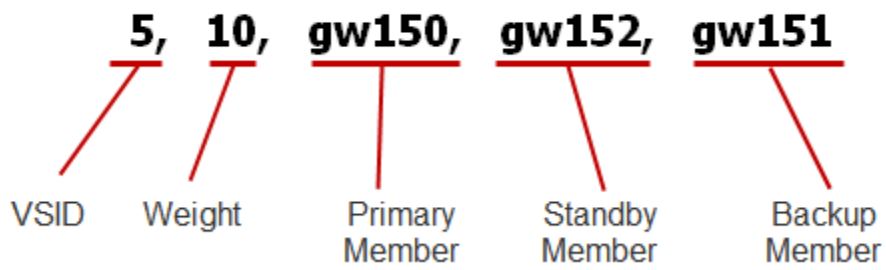
You can use the VSLS configuration file to change member priorities for Virtual Systems after adding or removing a member.

VSLS Configuration File

The VSLS configuration file is a comma separated value (CSV) text file that contains configuration settings for all Virtual Systems controlled by a management server. All lines preceded by the # symbol are comments and are not imported into the management database.

```
# Check Point VSX - VS Load Sharing configuration file
#
# Administrator         : aa
# SmartCenter/Main Domain Management Server : 192.168.50.160
# Generated on          : Thu Jul 23 13:08:42 2009
#
#
#
# VSID, Weight, Active member, Standby member, Backup member #1
# Virtual System name: vs1
2,10,gw150,gw151,gw152

# Virtual System name: vs2
3,10,gw151,gw152,gw150

# Virtual System name: vs3
4,10,gw152,gw150,gw151

# Virtual System name: vs4
6,10,gw151,gw150,gw152

# Virtual System name: vs5
5,10,gw150,gw152,gw151
```

The configuration file contains one line for each Virtual System, consisting of the following data as shown below:



Each line contain the VSID, the weight assigned the Virtual System, one primary member and one standby member. Additional backup members are listed following the standby member.

Exporting a VSLS configuration

The most common way to use VSLS configuration files is to initially define your cluster environment and virtual systems using SmartDashboard.

To export a VSLS configuration to a text file:

1. From the VSLS menu, select "**6. Export configuration to a file**".
2. Enter a file name, include its fully qualified path, for example:
   /home/admin/MyConfiguration

Processing Options

You can insert the following commands in the VSLS Configuration file to display audit trail information while validating and processing data. Each of the commands act as a toggle, whereby the first occurrence of a

command enables the action and the next occurrence disables it. These options his allow you to efficiently debug very long configuration files by displaying or logging only suspicious sections of the data.

Command	Action
!comments	Sequentially displays comment lines (those preceded with the '#' character) contained in the configuration file. You can insert comments into the configuration file to indicate which virtual systems are currently being processed or to provide status information as the parser processes the data.
!verbose	Displays whether or not each data line has been successfully verified and the configuration parameters for each Virtual System.
!log	Saves **!comments** and **!verbose** information in the **vsx_util.log** file.

Importing a VSLS configuration

To import a VSLS configuration from a text file:

1. From the VSLS menu, select "**5. Import configuration from a file**".
2. Enter the file name, include its fully qualified path, for example:
   /home/admin/MyConfiguration
3. At the "**Save & apply configuration ?**" prompt, enter "**y**" to continue.

During the import process, the parser reads the configuration file and attempts to validate the contents. Errors are displayed on the screen together with the offending line number. If either the **!comments** or **!verbose** processing options are enabled, the appropriate information appears on the screen.

The process update process may take several minutes or longer to complete, depending on the quantity of Virtual Systems, Domain Management Servers and cluster members.

Configuring Virtual Systems in Bridge Mode

This section presents configuration information and procedures for Virtual Systems in the Bridge mode. By implementing native layer-2 bridging instead of IP routing, you can add Virtual Systems without affecting the existing IP structure.

When in the Bridge mode, Virtual System interfaces do not require IP addresses. You may optionally assign an IP address to the Virtual System itself (not the interfaces) to enable layer-3 monitoring. This feature enhances network fault detection.

VSX supports the following Bridge mode models:

- **STP Bridge Mode**: Provides redundancy while preventing undesirable loops between redundant switches.
- **Active/Standby Bridge Mode**: Provides path redundancy and loop prevention, while offering seamless support for Virtual System Load Sharing and overcoming many of the limitations of STP.

Overview

STP Bridge Mode

This section presents the procedures for enabling and configuring the STP Bridge mode for Virtual Systems and VSX gateways.

Defining the Spanning Tree Structure

Define and configure the Spanning Tree structure according to your network requirements. Please refer to your hardware documentation for the specific procedures for your network deployment.

Enabling STP Bridge Mode when Creating Member

When creating a new VSX gateway for use as a cluster member, configure the following cluster options during the initial configuration process (*sysconfig* or *cpconfig*):

1. Enter '`y`' in response to the "`Would you like to install a Check Point clustering product?` question".
2. Enter '`n`', if prompted, to disable the **Active/Standby Bridge mode**. Continue with the remainder of the initial configuration process.

Enabling the STP Bridge Mode for Existing Members

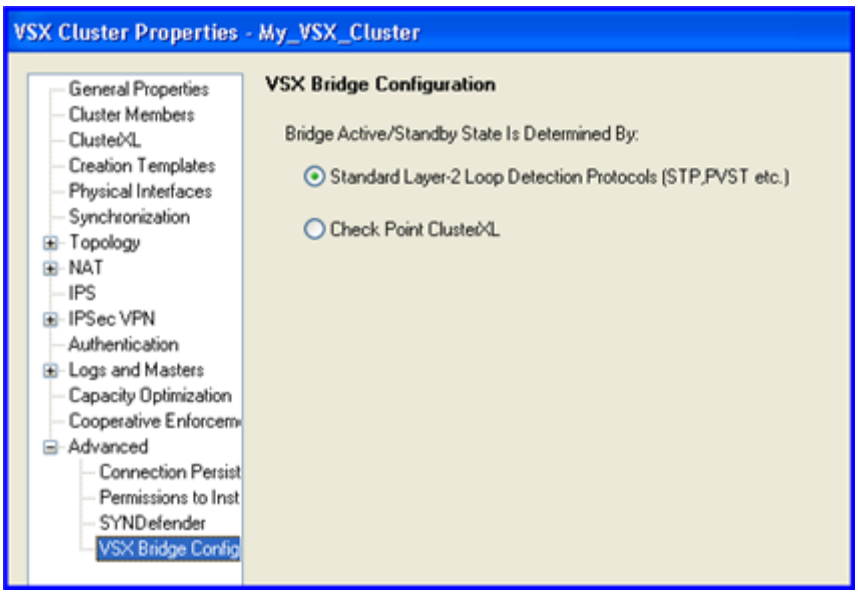
To enable the STP Bridge mode for existing cluster members:

1. Execute the `cpconfig` command.
2. Enable **cluster membership for this member** (If a numerical value appears here, cluster membership has already been enabled).
3. Disable **ClusterXL for Bridge Active/Standby**.
4. Reboot the member.

Configuring Clusters for STP Bridge Mode

To enable the STP Bridge mode for a cluster:

1. Open the desired cluster definition in SmartDashboard and navigate to **VSX> Cluster Properties > Advanced > VSX> Bridge Configuration**.



2. Enable the **Standard Layer-2 Loop Detection Protocols** option.

Configuring Virtual Systems for the STP Bridge Mode

To configure a Virtual System to use the Bridge mode, you must define it as a Virtual System in the Bridge mode when initially creating it. You cannot reconfigure an existing, non-Bridge mode Virtual System to use the Bridge mode at a later time.

Configuring PVST+ Load Sharing

Defining the Spanning Tree Structure

Define and configure the Spanning Tree structure for each VLAN according to your network deployment. Please refer to your network hardware documentation for specific procedures.

Configuring a Cluster for PVST+ Load Sharing

To configure a VSX cluster for PVST+ load sharing, perform the procedures described in the STP Bridge Mode section ("STP Bridge Mode" on page 119).

Active/Standby Bridge Mode

This section presents the procedures for enabling and configuring the Active/Standby Bridge mode for Virtual Systems and VSX gateways.

Enabling and Configuring Active/Standby Bridge Mode

Enabling Active/Standby Bridge Mode for a New Member

When creating a new cluster member, enable the following cluster options during the initial configuration (*sysconfig or cpconfig*) process:

1. Enter '`y`' in response to the "`Would you like to install a Check Point clustering product?` question".
2. If you enable the **Per Virtual System State** feature, which is required for VSLS, the Active/Standby Bridge mode is enabled automatically, and no further action is necessary.
3. If you elected not to enable VSLS, an option to enable the **Active/Standby Bridge mode** feature appears. Enter '`y`' and continue with the remainder of the gateway initial configuration process.

Enabling Active/Standby Bridge Mode for Existing Members

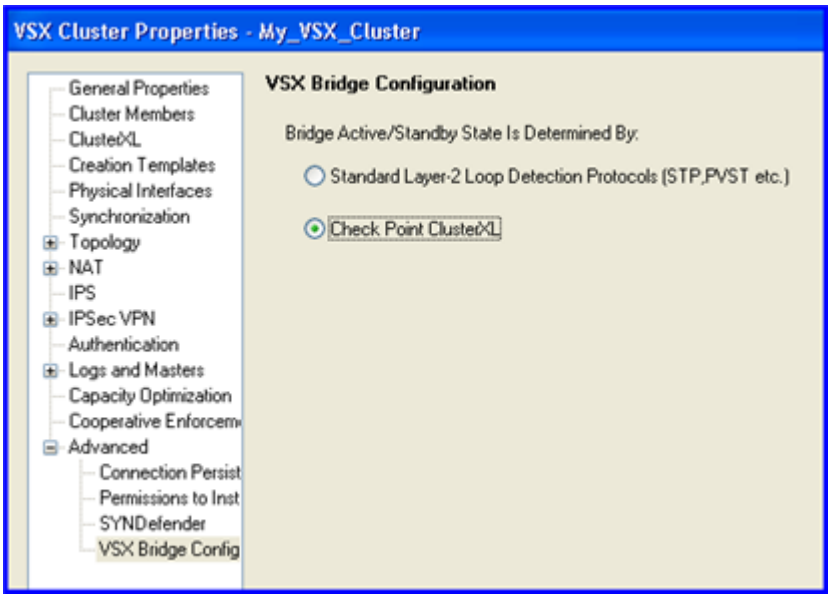
To enable the Active/Standby Bridge mode on existing Virtual Systems:

1. Execute the `cpconfig` command.
2. Enable ClusterXL for Bridge Active/Standby
3. Reboot the member

Configuring Clusters for Active/Standby Bridge Mode

To enable the Active/Standby Bridge mode for a cluster:

1. Open the cluster definition in SmartDashboard and navigate to **VSX Cluster Properties > Advanced > VSX Bridge Configuration**.



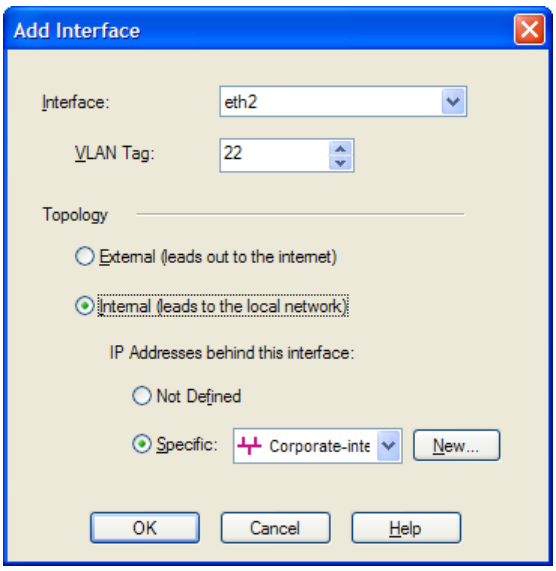
2. Enable the **Check Point ClusterXL** option, which enables the Active/Standby Bridge mode loop detection algorithms contained in ClusterXL.

Configuring Virtual Systems for Active/Standby Bridge Mode

To configure a Virtual System to use the Bridge mode, you must define it as a Virtual System in the Bridge mode when initially creating it. You cannot reconfigure an existing, non-Bridge mode Virtual System to use the Bridge mode at a later time.

To configure a Virtual System for the Active/Standby Bridge mode:

1. On the **Virtual System Wizard** - **General Properties** page, enable the Bridge mode option.
2. On the **Network Configuration** page, Click **Add** to open the **Add Interface** window. Define an internal and external interface as follows:



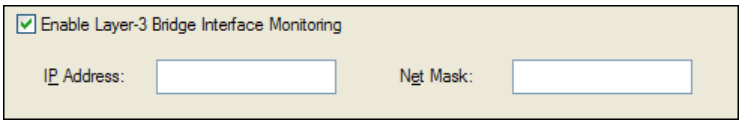
   a) Select an interface from the list.

   b) Enter a unique VLAN tag.

   c) Select either an internal or external interface.

   d) For internal interfaces, optionally select an IP address or Network that connects to the interface.

3. On the **Network Configuration** page, optionally, enable Layer-3 bridge interface monitoring. This feature assists in detecting network faults for failover. The IP address must be unique and be located on the same subnet as the protected network.



Advanced Clustering Configuration

This section presents several advanced cluster scenarios and procedures for their configuration.

Clusters on the Same Layer-2 Segment

The recommended cluster architecture contains interfaces connect to a Layer-2 segment that is isolated from other clusters. When configuring a cluster with only two members, you should connect the secured interfaces of the sync network using a crossover cable.

However, in a deployment where multiple clusters need to connect to the same Layer-2 segment, the same MAC address may be used by more than one cluster for Cluster Control Protocol (CCP) communication. This may direct traffic to the incorrect cluster. In this case you will need to modify the source MAC address(es) of the clusters.

This section describes how source MAC addresses are assigned, and explains how to change them. This procedure applies to both ClusterXL and OPSEC certified clustering products using the High Availability mode.

Source Cluster MAC Addresses

Cluster members use CCP to communicate with each other. In order to distinguish CCP packets from ordinary network traffic, CCP packets are given a unique source MAC address.

- The first four bytes of the source MAC address are all zero: 00.00.00.00
- The fifth byte of the source MAC address is a "magic" number, a number that encodes critical information in a way intended to be opaque. Its value indicates its purpose:

Default Value Of Fifth Byte	Purpose
`0xfe`	CCP traffic
`0xfd`	Forwarding layer traffic

- The sixth byte is the ID of the source cluster member

When multiple clusters are connected to the same Layer-2 segment, setting a unique value to the fifth byte of the MAC source address of each cluster allows them to coexist on the same Layer-2 segment.

Changing a Cluster's MAC Source Address

To change a cluster's MAC source address, run the following commands on each cluster member:

```
fw ctl set int fwha_mac_magic <value>
fw ctl set int fwha_mac_forward_magic <value>
```

The default values of the parameters `fwha_mac_magic` and `fwha_mac_forward_magic` appear in the following table:

Parameter	Default value
`fwha_mac_magic`	`0xfe`
`fwha_mac_forward_magic`	`0xfd`

Use any value as long as the two gateway configuration parameters are different. To avoid confusion, do not use the value 0x00.

Making the Change Permanent

You can configure the above configuration parameters to persist following reboot. For SecurePlatform machines:

1. Use a text editor to open the file `fwkern.conf`, located at `$FWDIR/boot/modules/`.
2. Add the line `Parameter=<value in hex>`. Make sure there are no spaces.

Monitoring all VLANs with ClusterXL

By default, ClusterXL only monitors two VLANS for failure detection and failover. These are the highest and lowest VLAN tags defined for a given interface.

For example, if the topology for interface eth1 includes several VLAN tags in the range of eth1.10 to eth1.50, ClusterXL only monitors VLANs eth1.10 and eth1.50 for failure. Failures on any of the other VLANs are not detected in the default configuration.

Note - The command line option `cphaprob -a if` displays the highest and lowest VLANs being monitored.

When both the highest and lowest VLANs fail, all the VLANs are considered down, and a failover occurs. This means that if a VLAN which is *not* listed as the highest or lowest goes down, the trunk is still considered "up", and no failover occurs.

There are instances in which it would be advantageous to monitor all the VLANs in the trunk, not just the highest and lowest, and initiate a failover when any one of the VLANs goes down.

To enable monitoring of all VLANs, enable the `fwha_monitor_all_vlans` property in `$FWDIR/boot/modules/fwkern.conf`.

Note - Monitoring all VLANS is enabled automatically when the Per VLAN state option is enabled.

Enabling Dynamic Routing Protocols

ClusterXL supports Dynamic Routing (Unicast and Multicast) protocols as an integral part of the SecurePlatform VSX installation. As the network infrastructure views the clustered gateway as a single logical entity, failure of a cluster member will be transparent to the network infrastructure and will not result in a ripple effect.

Components of the System

Virtual IP Integration

All cluster members use the cluster IP address.

Routing Table Synchronization

Routing information is synchronized among the cluster members using the Forwarding Information Base (FIB) Manager process. This is done to prevent traffic interruption in case of failover, and to support VSLS. The FIB Manager is the responsible for the routing information.

Failure Recovery

Dynamic Routing on ClusterXL avoids creating a ripple effect upon failover by informing the neighboring routers that the router has exited a maintenance mode. The neighboring routers then reestablish their relationships to the cluster, without informing the other routers in the network. These restart protocols are widely adopted by all major networking vendors.

The following table lists the RFC and drafts compliant with Check Point Dynamic Routing:

Protocol	RFC or Draft
OSPF LLS	draft-ietf-ospf-lls-00
OSPF Graceful restart	RFC 3623
BGP Graceful restart	draft-ietf-idr-restart-08

Dynamic Routing in ClusterXL

The components listed above function "behind-the-scenes." When configuring Dynamic Routing on ClusterXL, the routing protocols automatically relate to the cluster as they would to a single device.

When configuring the routing protocols for each cluster member, define each member identically, and use the cluster IP address (not the member's physical IP address). In the case of OSPF, the router ID must be defined and identical on each cluster member. When configuring OSPF restart, you must define the restart type as `signaled` or `graceful`. For Cisco devices, use type `signaled`.

Use command line interface in SecurePlatform to configure each cluster member. The table below is an example of the proper syntax for cluster member A.

Enabling OSPF on cluster member A

```
--------- Launch the Dynamic Routing Module
vsx1:0]# router
ER0 999 Unable to connect to host 'localhost'!
ER0 999 Dynamic Routing is not supported on VSX gateway/cluster
Use 'vrf-connect' to enter specific Virtual System
(disconnected)>vrf-connect 1
localhost.localdomain- VRF-1>enable
localhost.localdomain- VRF-1#configure terminal

--------- Enable OSPF and provide an OSPF router ID
localhost.localdomain- VRF-1(config)#router ospf 1
localhost.localdomain- VRF-1(config-router-ospf)#router-id
192.168.116.10
localhost.localdomain- VRF-1(config-router-ospf)#restart-type [
graceful | signaled ]
localhost.localdomain- VRF-1(config-router-ospf)#redistribute
kernel

--------- Define interfaces/IP addresses on which OSPF runs (Use
the cluster IP
localhost.localdomain- VRF-1(config-router-ospf)#network
1.1.10.10  0.0.0.0 area 0.0.0.0
localhost.localdomain- VRF-1(config-router-ospf)#network
1.1.10.20  0.0.0.0 area 0.0.0.0

-------- Exit the Dynamic Routing Module
localhost.localdomain- VRF-1(config-router-ospf)#exit
localhost.localdomain- VRF-1(config)#exit

-------- Write configuration to disk
localhost.localdomain- VRF-1#write memory
IU0 999 Configuration written to '/etc/gated1.ami'
localhost.localdomain- VRF-1#\
```

The same configuration needs to be applied to each cluster member.

As the FIB Manager uses TCP 2010 for routing information synchronization, the Security Policy must accept TCP 2010 to and from all cluster members. This is enabled as an implied rule.

Chapter 7

Working with URL Filtering

In This Chapter

Introduction	126
Configuring URL Filtering	127

Introduction

Access to the Internet can expose your organization to a variety of security threats and negatively affect employee productivity as a result of non-work-related surfing and downloading of files. Due to problems associated with employee web surfing, organizations are turning to Web Filtering to control employee Internet access, reduce legal liability and improve organizational security. Web Filtering enforces filtering rules based on the organization's needs and predefined categories made up of URLs and patterns of URLs.

Web Filtering includes reporting and monitoring tools that capture and present web traffic data, providing organizations with an in-depth look at how web surfing affects their organization's security and supports decisions regarding web surfing limitations.

A web filter is a function that screens incoming web pages to determine whether or not to display their web content. The web filter verifies the web page URL against a list of approved sites and blocks access to complete sites or pages within sites that contain objectionable material (for example, pornography, illegal software and spyware).

Terminology

The following terms are used in Web Filtering applications:

- **Allow List**: A list of allowed URL addresses, for example, a URL in the Allow List is allowed even if it is associated with a category that is blocked.
- **Block List**: A list of blocked URL addresses, for example, a URL in the Block List is blocked even if it is associated with a category that is not blocked.
- **Blocking Notifications**: Contains the message that appears when a URL address is blocked and the URL to which a blocked URL address is redirected.
- **Category**: Contains a group of topics sharing a common attribute (for example, crime, education and games.
- **Network Exceptions**: Contains a list of connections for which Web Filtering should not be enforced.
- **Web Filter**: Allows you to allow or block URLs based on network connections and/or an external categorized database and local exception lists.

Functional Overview

When incoming Web traffic arrives at a gateway on which Web Filtering is active, it first applies the general security policy rules. Traffic allowed by these rules is then inspected for permitted URLs and/or IP addresses.

The Web filtering engine inspects incoming requests and assigns one or more filter categories according to information contained in the Content Inspection database. The Content Inspection database is updated periodically to ensure accurate category assignment.

Following category assignment, the Web Filtering engine then blocks or allows the traffic according to one or more of the following rule types:

- **Network Exceptions**: Override Web Filtering rules based on predefined combinations of source and destination locations.
- **Address Rules**: Traffic destined for specific URLs/IP addresses is either blocked or allowed according to a Blocked List (blacklist) or an Allowed List (white list). These lists override decisions based on categories.
- **Category Rules**: Traffic is blocked or allowed according to the category assigned to it.

For example, if a URL or IP address matches two or more categories, and one of those categories is blocked, the traffic is blocked. However, if the same address appears in the Allowed List or in a Network Exception definition, the traffic is allowed.

Configuring URL Filtering

The following procedure describes the process for enabling URL Filtering on VSX objects. It is important to note that you must configure **all** VSX cluster and VSX gateway objects before configuring virtual system objects.

The process consists of two basic procedures:

1. Enabling URL Filtering on VSX objects (clusters, VSX gateways and Virtual Systems).
2. Defining the URL Filtering Policy and installing it on the appropriate objects.

Enabling URL Filtering

To enable URL Filtering, perform the following steps on each individual VSX object. You should enable URL Filtering on all VSX objects before Virtual Systems.

1. In SmartDashboard, select a VSX object and open its property definition window.
2. On the **General Properties** page, select **URL Filtering** in the **Check Point Products List** to activate URL Filtering for that specific VSX object.
3. Install the Policy.
4. Repeat these steps for each VSX cluster, VSX gateway, and Virtual System.

Defining the URL Filtering Policy

Perform the following steps to define the URL Filtering policy in SmartDashboard.

1. On the SmartDashboard **Content Inspection** tab, select **URL Filtering > URL Filtering Policy**. The URL Filtering Policy window opens.
2. On the **URL Filtering** page, configure the following:
   a) In the **Category Selection** list, select the URL categories to block.
      - A green icon indicates that URLs associated with this category are allowed.
      - A red icon indicates that URLs associated with this category are blocked.
   b) In **Track Configuration**, select how to track a detected URL address. All options other than **None** generate a log record in SmartView Tracker.
3. Select **Advanced > Allow List** to add a URL and/or IP address to be allowed even if it is associated with a blocked category.
4. Select **Advanced > Block List** to add a URL and/or IP address to be blocked even if it is associated with an allowed category.

Note - The URL database also includes IP addresses. By Default, all IP addresses are allowed, even if included in the **Allow** or **Block** lists.

To enable the Allow and Block lists to work with IP addresses, use the GuiDBedit
utility and change the `categorize_http_request_method` parameter to `host_dns_and_ip` (the default value is `host_dn`s).

When defining IP addresses in the Allow or Block lists, you must append the '/' character to the end of each address string. If you fail to do this, the parser treats the entire IP address as a wildcard prefix and may inappropriately block or allow other IP addresses.

5. In the **Advanced** branch, select **Network Exceptions** to create a list of the network connections through which traffic should not be inspected or in order to enforce URL Filtering on all Web traffic. **Network Exceptions** works according to a source and destination Rule Base and overrides the URL Filtering engine.
6. In the **Advanced** branch, select one of the following **Blocking Notifications** in order to notify the user when the URL request is blocked:
   - Enter the message to be displayed when a URL is blocked according to the URL Filtering Policy.

   Or
   - Enter a URL to which the user is to be redirected.
7. Install the Policy on appropriate VSX objects.

Updating the Content Inspection Database

In order to ensure that URL Filtering protection is current and accurate, it is essential to update the Content Inspection database on a regular basis. The following database update methods are available:

- **Automatic Updates:** Updates occur automatically on a fixed schedule, according to predefined parameters.
- **Manual Updates**: You can update the database manually at any time according to parameters defined in a wizard.

Updates are available from the Check Point website. Prior to downloading, verify that:

- HTTP and HTTPS Internet connectivity with DNS is properly configured.
- You have a valid Check Point User Center user name and password.

Note -

- Database updates are performed using the VSX gateway management IP address.
- The first update may take several minutes, depending on your network bandwidth. Subsequent updates will take significantly less time because only incremental information is downloaded.

URL Filtering begins to work only after you perform the update and install the policy.

Configuring Automatic Updates

To configure automatic updates:

1. On the **Database Updates** page in the SmartDashboard **Content Inspection** tab, select the **Enable automatic updates** option.
2. Click **Configure Automatic Updates**.
   The **Automatic Updates** wizard opens.
3. Enter your Check Point User Center email address and password in the appropriate fields.
4. On the **URL Filtering** tab, select the update frequency (in hours).
5. Configure **Tracking Configuration** parameters as appropriate.
6. Click **OK** to complete the definition.

Performing Manual Updates

To perform a manual database update:

1. On the **Database Updates** page in the SmartDashboard **Content Inspection** tab, click **Update databases now.** The **Update Databases** wizard opens.
2. In the first window, enter your Check Point User Center email address and password in the appropriate fields.
3. In the second window, select **Custom update**.
4. In the third window, select **URL Filtering**.
5. In the final window select the VSX gateway and click **Add > Do not select any other objects**.
6. Click **Finish** to perform the update.

Password Bypass

Password bypass is a feature that allows authorized users to override Web Filter traffic rejections by entering a password. Users can be granted access to a blocked domain, and all links pointing to locations within that domain, for a limited amount of time that is configured by the administrator.

This allows administrators to exempt specific users from URL Filtering limitations and to allow these users to bypass URL Filtering false positives. For example, a network security officer might be granted access to hacking or identity theft sites in order to develop countermeasures.

Using Password Bypass

Upon notification that access to a specific Website has been blocked by URL Filtering, users receive an opportunity to enter their network password to override the blockage. After entering the correct password, access is granted to any location on the blocked domain for a period of time configured in the SmartDashboard Bypass options. After the specified amount of time, the user is required to re-enter the password to regain access.

Configuring Password Bypass

To allow users to use password bypass, perform the following steps:

1. In SmartDashboard, create a user group with the name `UF_bypass`.
2. Add those specific users or user groups who have password override privileges to this group.
3. Install the Policy.

URL Filtering Acceleration

A single Web page display often contains numerous HTTP requests in order to display all of it's components. Normally, each individual request is inspected by the URL Filtering engine, resulting in degraded performance. The URL Filtering acceleration option improves performance by inspecting only the first request for a given page and applying its decision to all subsequent requests.

You can enable or disable the acceleration feature via the command line. Acceleration is enabled by default:

To enable URL Filtering acceleration, run the following command:

```
fw ctl set int http_ufp_allow_acceleration 1
```

To disable URL Filtering acceleration, run the following command:

```
fw ctl set int http_ufp_allow_acceleration 0
```

To ensure that this setting remains in place after rebooting, add the following line to the `$FWIR/boot/module/fwkern.conf` file:

```
http_ufp_allow_acceleration=0
```

Chapter 8

Working with Link Aggregation

In This Chapter

Link Aggregation Overview	130
Configuring Link Aggregation for High Availability	134
Link Aggregation - Load Sharing Mode	137
Changing the Bond Interface Mode	145
Enslaving Interfaces to a Bond	145
Detaching Interfaces from a Bond	146
Deleting a Bond	146
Changing an Existing Interface to a Bond	148
Troubleshooting Bonded Interfaces	148

Link Aggregation Overview

Link aggregation, also known as interface bonding, joins multiple physical interfaces together into a virtual interface, known as a **bond interface**. A bond interface can be configured for High Availability redundancy or for load sharing, which increases connection throughput above that which is possible using one physical interface.

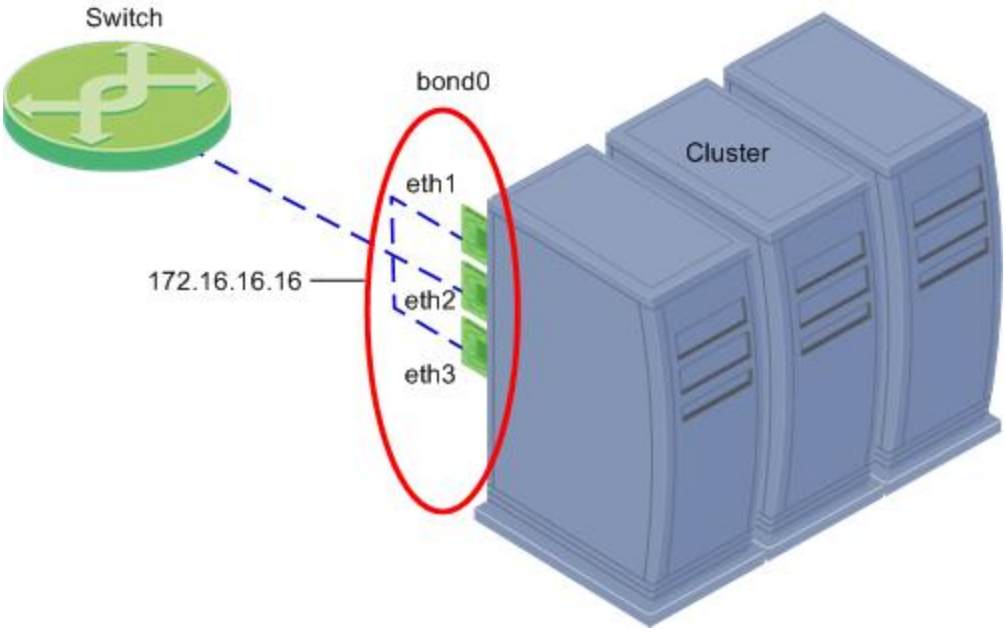
Link Aggregation Terminology

- **Link Aggregation (Interface Bonding)**: Networking technology that binds multiple physical interfaces together into one virtual interface.
- **Bond**: A group of physical interfaces that operate together as one virtual interface and share an IP address and MAC address. A bond is identified by the cluster by its **Bond ID** (for example: *bond0*).
- **Bond Interface**: The interface assigned to the bond itself.
- **Slave (enslaved interface)**: A physical interface that is a member of a bond. Slaves do not have an IP Address and share the same MAC address.

How Link Aggregation Works

A bond contains a minimum of one and may contain up to eight slave interfaces. All slave interfaces contained in a bond share a common IP and MAC address. We recommend that each cluster member contain the same quantity of identical slave interfaces.

Figure 8-29 Bond with three slave interfaces



Note - Link Aggregation is only supported on Check Point SecurePlatform machines.

You can configure Link Aggregation using one of the following strategies:

- **High Availability (Active/Backup)**: Ensures redundancy in the event of interface or link failure. This option also provides switch redundancy.
- **Load Sharing (Active/Active)**: All interfaces are active, but handle different connections simultaneously. Traffic is balanced amongst slave interfaces to maximize throughput. The Load Sharing option does not support switch redundancy.

High Availability Overview

Clusters, by definition, provide redundancy and high availability at the Security Gateway level. Link Aggregation, however, adds interface and switch redundancy by providing automatic failover to a standby interface card within the same member.

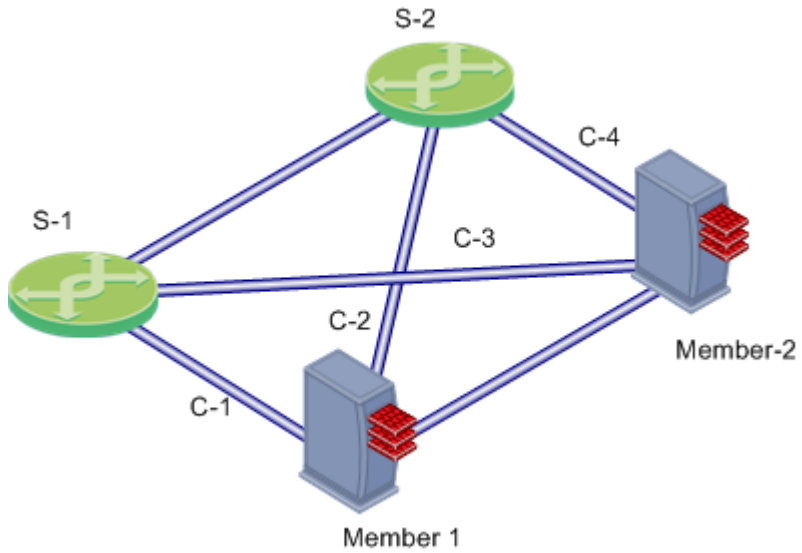
In a High Availability deployment, only one interface is active at a time. If an interface or connection fails, the bond fails over to a standby slave interface. Failover occurs in one of the following cases:

- An active interface detects a link state failure in a monitored interface
- ClusterXL detects a failure in sending or receiving Cluster Control Protocol (CCP) keep-alive packets

Fully Meshed Redundancy via Interface Bonding

The Link Aggregation High Availability mode, when deployed with ClusterXL, enables a higher level of reliability by providing granular redundancy in the network. This granular redundancy is achieved by using a fully meshed topology, which provides for independent backups for both NICs and switches.

The following figure illustrates this concept.



In this scenario:

- Member-1 and Member-2 are cluster members in the High Availability mode
- S-1 and S-2 are switches
- C-1, C-2, C-3 and C-4 are network connections

Load Sharing Overview

Load sharing provides the ability to spread traffic over multiple slave interfaces, in addition to providing interface redundancy. All interfaces are always active. While Load Sharing has the advantage of increasing throughput, it requires connecting all interfaces to a single switch and cannot provide switch redundancy.

Traffic is balanced between interfaces in a manner similar to the way load sharing balances traffic between cluster members. Load sharing operates according to either the IEEE 802.3ad or the XOR standard.

In Load Sharing mode, each individual connection is assigned to a specific slave interface. For a specific connection, only the designated slave interface is active. In the event of a failure of the designated slave interface, the traffic fails over to another interface, adding that connection's to the traffic it is already handling.

Bond Failover

Either of the following failure scenarios can induce bond failover:

- An active interface detects a link state failure in another monitored interface
- ClusterXL detects a failure in sending or receiving Cluster Control Protocol (CCP) keep-alive packets

Either of these occurrences will induce a failover, either to another slave interface within the bond, or between cluster members, depending on the circumstances. Refer to *Configuring Failover Mode* in the *R75.20 ClusterXL Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12265) for details.

The following sections describe the types of failover processes.

Note - The bond failover operation requires a network interface card that supports the Media-Independent Interface (MII) standard.

Link State Initiated Failover

Link-state initiated failover occurs in the following sequence:

1. The active slave interface detects a **down** link state.

2. The bond initiates failover to a standby interface. Since this is a failover within the bond, the status of the other cluster member is unaffected.
3. If the standby interface continues to detect a link failure, and the initial interface is still down, failover to other cluster members occurs.

CCP Initiated Failover

CCP failover occurs only when other cluster members are not down, in the following sequence.

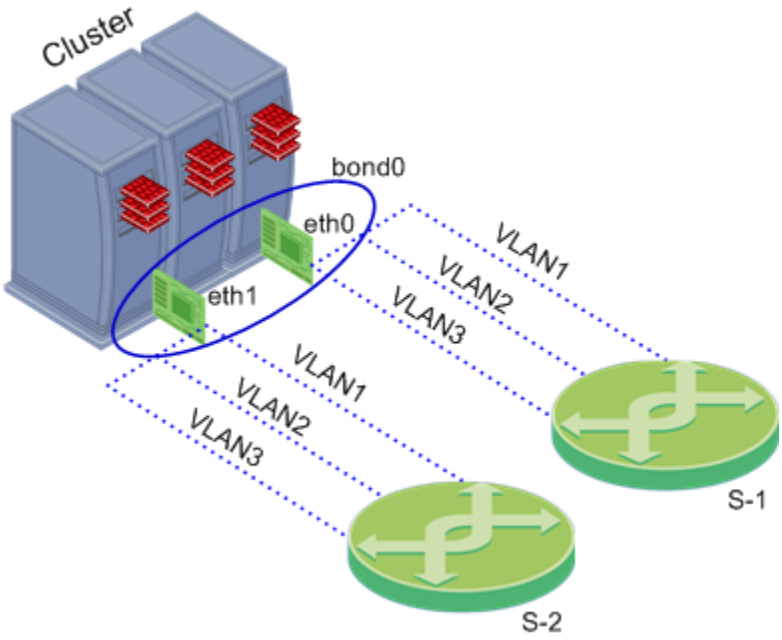
1. ClusterXL detects a problem sending or receiving of CCP packets.
2. ClusterXL initiates an internal bond failover.
3. ClusterXL monitors CCP packet transmission and reception. If additional problems are detected within three minutes, the system initiates a failover to another cluster member.

Failover Support for VLANs

ClusterXL monitors VLAN IDs for connectivity failure or miscommunication, and initiates failover when necessary. By default, both the highest and the lowest VLAN IDs are monitored for failure. This is done by sending ClusterXL Control Protocol (CCP) packets on round-trip paths at a set interval. The following figure illustrates an interface bond supporting multiple VLANs.

You can optionally configure VSX to Monitor all VLANs.

Figure 8-30 Interface Bonding with VLANS



When a failure is detected, a log of the failure is recorded in SmartView Tracker.

Monitoring the Highest and Lowest VLAN IDs

By default, the highest and lowest VLAN IDs indicate the status of the physical connection. these VLAN IDs are always monitored and a connectivity failure in either initiates a failover. In most deployments this is the desired setting, as it supports the primary purpose of the feature (detecting a connectivity failure) and the traffic generated on the network is light. However, this setting will not detect a VLAN configuration problem on the switch.

Bond Interface & Interface Limitations

- You can define a maximum of 4096 interfaces on a SecurePlatform machine. The total number of bond interfaces in use is the sum of bonds plus the number of slave interfaces contained in each bond.

- Up to eight interfaces can be defined in a Link Aggregation deployment.

Configuring Link Aggregation for High Availability

This section explains how to create a new High Availability Link Aggregation deployment. A new deployment contains no VSX gateways, cluster objects or Multi-Domain Security Management Domains. Do these procedures in sequence:

Deployment Tasks

Creating a New Deployment 134
Upgrading an Existing Deployment 135

Creating a New Deployment

Deployment Tasks:

Defining the Interface Bond 134
Defining Slave Interfaces as Disconnected 134
Verifying that the Bond is Functioning Properly 135
Creating the Cluster. 135

Defining the Interface Bond

When the slave interfaces are without IP addresses, define the bond:

1. Start the SecurePlatform configuration utility:
   `sysconfig`
2. Select `Network Connections`.
3. Select `Add new connection`.
4. Select `Bond`.
5. For each interface to be enslaved under the bond, type its number in the list, and press Enter.
6. Enter `n` to go to the next step.
7. Select `High Availability`.
8. Choose whether to use default parameters (recommended) or to customize them.
9. Choose whether to set a primary slave interface, or not (recommended).
   A primary slave interface, after failing and coming back up, automatically returns to Active status, even if failover to the other interface occurred. If there is no primary interface, failover causes the other interface to become active and remain so until it fails.
10. Define the IP address and network mask of the new interface bond.
11. Exit the SecurePlatform configuration utility.

Defining Slave Interfaces as Disconnected

In a bond, slave interfaces should be configured as *disconnected*. Disconnected interfaces are cluster member interfaces that are not monitored by the ClusterXL mechanism. If a disconnected interface fails, failover does not occur.

To define a slave interface as disconnected in SecurePlatform:

1. On the cluster member machine, open the file named **discntd.if** in the directory **$FWDIR/conf/** in a text editor. If this file does not yet exist, you need to create it.
2. Enter the name of each physical interface contained in the bond on a separate line, as shown in the following example:

```
pimreg
eth5
eth6
```

3. Repeat this process for each member.

Verifying that the Bond is Functioning Properly

After installation or failover, it is recommended to verify that the bond is up, by displaying bond information.

1. Run:
   `cphaprob -a if`
   Make sure that the bond status is reported as `UP`.
2. Run:
   `cphaconf show_bond <bond name>`
   Check that the bond is correctly configured.

Creating the Cluster.

Define the cluster object ("Creating a New Cluster" on page 93) using SmartDashboard. During the cluster definition process, SmartDashboard automatically fetches the topology from the cluster members, including the newly defined bond interfaces.

Upgrading an Existing Deployment

This section presents the procedures for upgrading an existing deployment to use Link Aggregation High Availability. An existing deployment is one that contains previously defined cluster objects on existing hardware platforms. You perform most procedures, including defining VSX clusters objects and policies, using SmartDashboard.

The process for converting an existing deployment to Link Aggregation High Availability consists of the following procedures:

Upgrade Tasks:

Removing IP Addresses from Slave Interfaces 135
Defining the Interface Bond 136
Defining Slave Interfaces as Disconnected 136
Verifying that the Bond is Functioning Properly 136
Reconfiguring Topology 136

Removing IP Addresses from Slave Interfaces

Before you define an interface bond, make sure the slave (physical) interfaces do not have IP addresses:

1. Start the SecurePlatform configuration utility:
   `sysconfig`
2. Select `Network Connections`.
3. For each interface to be enslaved:
   a) Select `Configure connection`.
   b) Select the relevant physical interface.
   c) Select `Remove IP from interface`.
   d) Return to Network Connections.
4. Exit the SecurePlatform configuration utility.

Defining the Interface Bond

When the slave interfaces are without IP addresses, define the bond:

1. Start the SecurePlatform configuration utility:
   `sysconfig`
2. Select `Network Connections`.
3. Select `Add new connection`.
4. Select `Bond`.
5. For each interface to be enslaved under the bond, type its number in the list, and press Enter.
6. Enter `n` to go to the next step.
7. Select `High Availability`.
8. Choose whether to use default parameters (recommended) or to customize them.
9. Choose whether to set a primary slave interface, or not (recommended).
   A primary slave interface, after failing and coming back up, automatically returns to Active status, even if failover to the other interface occurred. If there is no primary interface, failover causes the other interface to become active and remain so until it fails.
10. Define the IP address and network mask of the new interface bond.
11. Exit the SecurePlatform configuration utility.

Defining Slave Interfaces as Disconnected

In a bond, slave interfaces should be configured as *disconnected*. Disconnected interfaces are cluster member interfaces that are not monitored by the ClusterXL mechanism. If a disconnected interface fails, failover does not occur.

To define a slave interface as disconnected in SecurePlatform:

1. On the cluster member machine, open the file named **discntd.if** in the directory **$FWDIR/conf/** in a text editor. If this file does not yet exist, you need to create it.
2. Enter the name of each physical interface contained in the bond on a separate line, as shown in the following example:

```
pimreg
eth5
eth6
```

3. Repeat this process for each member.

Verifying that the Bond is Functioning Properly

After installation or failover, it is recommended to verify that the bond is up, by displaying bond information.

1. Execute the following command:
   `cphaprob -a if`
   Check that the bond status is reported as `UP,` using the `cphaprob -a if` command.
2. `Execute the following command:`
   `cphaconf show_bond <bond name>`
   Check that the bond is correctly configured, using the `cphaconf show_bond` command.

Reconfiguring Topology

At this point, you need to reconfigure the relevant objects to connect to the newly created bond. This includes Virtual Systems, Virtual Routers and Virtual Switches. You can perform these actions using SmartDashboard. In most cases, these definitions can be found in the object **Properties** window.

For large existing VSX deployments containing many Domain Management Servers and virtual devices, use the *vsx_util change_interfaces* command to reconfigure existing object topologies. For example, in a Multi-Domain Security Management deployment with 200 Domains, each with many virtual devices, it is much faster to use *vsx_util change_interfaces*. This command automatically replaces the interface with the new bond on all relevant objects.

Reconfiguring the Bond using SmartDashboard

To configure the newly created bond:

1. In the SmartDashboard navigation tree, double-click the VSX gateway or cluster object.
2. In the **Properties** window, select the **Physical Interfaces** branch.
3. Click **Add** to add the new Bond to the cluster object.
   a) In the **Physical Interface Properties** window, enter the bond name. This name must be exactly the same as the name assigned to the bond when it was created with the *sysconfig*.
   b) If the bond is a VLAN trunk, enable the **VLAN Trunk** option.
4. On the **Topology** page:
   a) Double-click the interface(s) to be replaced.
   b) Select the bond interface from the **Interface** list.
   c) Click OK to push the configuration.
5. Install the policy.

Reconfiguring Topology using vsx_util change_interfaces

To reconfigure objects using vsx_util change_interfaces:

Important - All Domain Management Servers must be unlocked in order for this operation to succeed.

1. Close SmartDashboard for all Multi-Domain Security Management Domain Management Servers using the designated interface.
2. On the management server, enter the Expert Mode and execute the *vsx_util change_interfaces* command.
3. Enter the Security Management Server, or Multi-Domain Security Management main Domain Management Server IP address.
4. Enter the administrator name and password as requested.
5. Enter the VSX cluster object name.
6. Select **Apply changes to the management database and to the VSX Gateway/Cluster members immediately**.
7. When prompted, select the interface to be replaced.
8. When prompted, select the replacement bond interface.
9. If you wish to replace additional interfaces, enter "**y**" when prompted and repeat steps 6 and 7.
10. To complete the process, enter "**n**".

Link Aggregation - Load Sharing Mode

In This Section

Creating a Bond in a New Deployment 138
Upgrading an Existing Deployment 141
Configuring Cisco Switches for Load Sharing 145

In Load Sharing mode, Link Aggregation supplies load sharing, in addition to High Availability. All slave interfaces are active, and connections are balanced between the bond's slave interfaces, similar to the way ClusterXL balances connections between cluster members.

In Load Sharing mode, each connection is assigned to a specific slave interface. For the individual connection, only one slave interface is active. On failure of that interface, the bond does failover of the connection to one of the other interfaces, which adds the failed interface's connection to the connections it is already handling.

Connections are balanced between slave interfaces according to network layers three and four, and follow one of these standards:

- 802.3ad - includes LACP and is the recommended mode, but some switches may not support this mode.
- XOR.

In Load Sharing mode, all the interfaces of a bond must be connected to the same switch. The switch itself must support and be configured for Link Aggregation, by the same standard (802.3ad or XOR) as the gateway bond.

Load Sharing needs Performance Pack to be running.

Creating a Bond in a New Deployment

This section presents procedures for creating a bond in a new VSX deployment. A new deployment is defined as one in which no VSX Gateways, cluster objects or Multi-Domain Security Management Domain Management Servers have been previously defined.

Note - You must also configure switches the 802.3ad standard.

Perform the following steps in sequence:

Creating Interface Bond in Load Sharing Mode	138
Defining Slave Interfaces as Disconnected	139
Setting Critical Required Interfaces	139
Setting Affinities	140
Verifying that the Bond is Functioning Properly	140
Creating the Cluster.	141

Creating Interface Bond in Load Sharing Mode

To create a new Interface Bond, perform the following steps on the VSX Gateway or all cluster members:

1. From VSX gateway or cluster member command line, run `sysconfig`.
2. Select `Network Connections`.
3. Select `Add new connection`.
4. Select `Bond`.
5. Select each slave interface to be included in the bond, by entering the number corresponding to the interface name.

```
Choose interfaces to be enslaved under the bond ('n' for next, 'e' to exit):
------------------------------------------------------------------
1) [_]eth0  3) [x]eth2  5) [_]eth4  7) [_]eth6
2) [x]eth1  4) [_]eth3  6) [_]eth5
------------------------------------------------------------------
(Note: configuration changes are automatically saved)
Your choice:
```

6. Type **n** to continue.
7. Select `Load Sharing`.
8. Choose the Load Sharing mode: 802.3ad or XOR.
9. Configure advanced parameters as follows:

   a) If you wish to accept the default advanced parameters (recommended for most installations) enter **1** and then press **n** to continue. Proceed to the next step. See below for the default values

b) Enter **2** and configure the following settings as required:

- **MII Monitoring Interval**: Specifies the MII link monitoring frequency in milliseconds. This determines how often the link state of each slave is inspected for link failures. A value of zero disables MII link monitoring. The default value of 100 ms is a good starting point.
- **Up Delay:** This option is only valid for MII link monitoring. Specifies the time, in ms, before enabling a slave interface upon link recovery. The value should be a multiple of the **MII Monitoring Interval** value. If not, it is automatically rounded down to the nearest multiple (default = 200 ms).
- **Down Delay**: This option is only valid for MII link monitoring. Specifies the time, in ms, before enabling a slave interface upon link failure. The value should be a multiple of the **MII Monitoring Interval** value. If not, it is automatically rounded down to the nearest multiple. (default = 200 ms)
- **LACP Rate**: Transmission frequency of LACPDU packets in 802.3ad mode (Default = Slow - 30 seconds).

10. To assign an IP address at this time. Enter **n** to continue. You assign the IP address using SmartDashboard at a later stage.

```
New connection 'bond0' has been created!


Do you want to assign an IP address to this bond? (y/n)
[y]: n
```

11. Repeat this procedure for each member.

Defining Slave Interfaces as Disconnected

In a bond, slave interfaces should be configured as *disconnected*. Disconnected interfaces are cluster member interfaces that are not monitored by the ClusterXL mechanism. If a disconnected interface fails, failover does not occur.

To define a slave interface as disconnected in SecurePlatform:

1. On the cluster member machine, open the file named **discntd.if** in the directory **$FWDIR/conf/** in a text editor. If this file does not yet exist, you need to create it.
2. Enter the name of each physical interface contained in the bond on a separate line, as shown in the following example:

```
pimreg
eth5
eth6
```

3. Repeat this process for each member.

Setting Critical Required Interfaces

A bond in Load Sharing mode is considered to be down when fewer than a critical minimum number of slave interfaces remain up.

When not explicitly defined, the critical minimum number of interfaces in a bond of n interfaces is n-1. Failure of a second interface will cause the entire bond to be considered down, even if the bond contains more than two interfaces.

If a smaller number of interfaces will be able to handle the expected traffic, you can increase redundancy by explicitly defining the number of critical interfaces. Divide your maximal expected traffic speed by the speed of your interfaces and round up to a whole number to determine an appropriate number of critical interfaces.

To explicitly define the number of critical interfaces, create and edit the following file:

```
$FWDIR/conf/cpha_bond_ls_config.conf
```

Each line of the file should be of the following syntax:

```
<bondname> <critical#>
```

For example, if bond0 has seven interfaces and bond1 has six interfaces, file contents could be:

```
bond0 5
```

```
bond1 3
```

In this case bond0 would be considered down when three of its interfaces have failed. bond1 would be considered down when four of its interfaces have failed.

Setting Affinities

If you are running Performance Pack in a multi-core system, after you define bonds, set affinities manually. Use the -s option of the `sim affinity` command (see *Performance Pack* documentation).

Note - sim affinity commands take effect only if the Performance Pack is enabled and actually running. Performance Pack begins running when you install a policy for the first time.

For optimal performance, set affinities according to the following guidelines:

1. Run `sim affinity` using the *-s* option.
2. Whenever possible, dedicate one processing core to each interface. See sk33520 (http://supportcontent.checkpoint.com/solutions?id=sk33250).
3. If there are more interfaces than cores, one or more cores handle two interfaces. Use interface pairs of the same position with internal and external bonds.

   a) To view interface positions in a bond, run:

   ```
   cat /proc/net/bonding/<bond name>.
   ```

   b) Note the sequence of the interfaces in the output, and compare this for the two bonds (external bond and its respective internal bond). Interfaces that appear in the same position in the two bonds are interface pairs and set to be handled by one processing core.

   For example, you might have four processing cores (0-3) and six interfaces (0-5), distributed among two bonds:

bond0	bond1
eth0	eth3
eth1	eth4
eth2	eth5

   Two of the cores will need to handle two interfaces each. An optimal configuration might be:

bond0		bond1	
eth0	core 0	eth3	core 0
eth1	core 1	eth4	core 1
eth2	core 2		
		eth5	core 3

Verifying that the Bond is Functioning Properly

After installation or failover, it is recommended to verify that the bond is up, by displaying bond information.

1. Run:
   ```
   cphaprob -a if
   ```
   Make sure that the bond status is reported as `UP`.
2. Run:
   ```
   cphaconf show_bond <bond name>
   ```
   Check that the bond is correctly configured.

Creating the Cluster.

Define the cluster object ("Creating a New Cluster" on page 93) using SmartDashboard. During the cluster definition process, SmartDashboard automatically fetches the topology from the cluster members, including the newly defined bond interfaces.

Upgrading an Existing Deployment

This section presents the procedures for upgrading an existing deployment to use Link Aggregation Load Sharing. An existing deployment is one that contains previously defined cluster objects on existing hardware platforms. You perform most procedures, including defining clusters objects and policies, using SmartDashboard.

The process for converting an existing deployment to Link Aggregation Load Sharing consists of the following procedures:

Tasks:

Removing IP Addresses from Slave Interfaces 141
Creating Interface Bond in Load Sharing Mode 141
Defining Slave Interfaces as Disconnected 142
Setting Critical Required Interfaces 142
Setting Affinities 143
Verifying that the Bond is Functioning Properly 144
Reconfiguring Topology 144

Removing IP Addresses from Slave Interfaces

Before you define an interface bond, make sure the slave (physical) interfaces do not have IP addresses:

1. Start the SecurePlatform configuration utility:
   `sysconfig`
2. Select `Network Connections`.
3. For each interface to be enslaved:
   a) Select `Configure connection`.
   b) Select the relevant physical interface.
   c) Select `Remove IP from interface`.
   d) Return to Network Connections.
4. Exit the SecurePlatform configuration utility.

Creating Interface Bond in Load Sharing Mode

To create a new Interface Bond, perform the following steps on the VSX Gateway or all cluster members:

1. From VSX gateway or cluster member command line, run `sysconfig`.
2. Select `Network Connections`.
3. Select `Add new connection`.
4. Select `Bond`.
5. Select each slave interface to be included in the bond, by entering the number corresponding to the interface name.

```
Choose interfaces to be enslaved under the bond ('n' for next, 'e' to exit):
------------------------------------------------------------------
1) [_]eth0  3) [x]eth2  5) [_]eth4  7) [_]eth6
```

```
2) [x]eth1  4) [_]eth3  6) [_]eth5
---------------------------------------------------------------
(Note: configuration changes are automatically saved)
Your choice:
```

6. Type **n** to continue.
7. Select `Load Sharing`.
8. Choose the Load Sharing mode: 802.3ad or XOR.
9. Configure advanced parameters as follows:
   a) If you wish to accept the default advanced parameters (recommended for most installations) enter **1** and then press **n** to continue. Proceed to the next step. See below for the default values
   b) Enter **2** and configure the following settings as required:
      - **MII Monitoring Interval**: Specifies the MII link monitoring frequency in milliseconds. This determines how often the link state of each slave is inspected for link failures. A value of zero disables MII link monitoring. The default value of 100 ms is a good starting point.
      - **Up Delay:** This option is only valid for MII link monitoring. Specifies the time, in ms, before enabling a slave interface upon link recovery. The value should be a multiple of the **MII Monitoring Interval** value. If not, it is automatically rounded down to the nearest multiple (default = 200 ms).
      - **Down Delay**: This option is only valid for MII link monitoring. Specifies the time, in ms, before enabling a slave interface upon link failure. The value should be a multiple of the **MII Monitoring Interval** value. If not, it is automatically rounded down to the nearest multiple. (default = 200 ms)
      - **LACP Rate**: Transmission frequency of LACPDU packets in 802.3ad mode (Default = Slow - 30 seconds).
10. To assign an IP address at this time. Enter **n** to continue. You assign the IP address using SmartDashboard at a later stage.

```
New connection 'bond0' has been created!


Do you want to assign an IP address to this bond? (y/n)
[y]: n
```

11. Repeat this procedure for each member.

Defining Slave Interfaces as Disconnected

In a bond, slave interfaces should be configured as *disconnected*. Disconnected interfaces are cluster member interfaces that are not monitored by the ClusterXL mechanism. If a disconnected interface fails, failover does not occur.

To define a slave interface as disconnected in SecurePlatform:

1. On the cluster member machine, open the file named **discntd.if** in the directory **$FWDIR/conf/** in a text editor. If this file does not yet exist, you need to create it.
2. Enter the name of each physical interface contained in the bond on a separate line, as shown in the following example:

```
pimreg
eth5
eth6
```

3. Repeat this process for each member.

Setting Critical Required Interfaces

A bond in Load Sharing mode is considered to be down when fewer than a critical minimum number of slave interfaces remain up.

When not explicitly defined, the critical minimum number of interfaces in a bond of n interfaces is n-1. Failure of a second interface will cause the entire bond to be considered down, even if the bond contains more than two interfaces.

If a smaller number of interfaces will be able to handle the expected traffic, you can increase redundancy by explicitly defining the number of critical interfaces. Divide your maximal expected traffic speed by the speed of your interfaces and round up to a whole number to determine an appropriate number of critical interfaces.

To explicitly define the number of critical interfaces, create and edit the following file:

```
$FWDIR/conf/cpha_bond_ls_config.conf
```

Each line of the file should be of the following syntax:

```
<bondname> <critical#>
```

For example, if bond0 has seven interfaces and bond1 has six interfaces, file contents could be:

```
bond0 5
bond1 3
```

In this case bond0 would be considered down when three of its interfaces have failed. bond1 would be considered down when four of its interfaces have failed.

Setting Affinities

If you are running Performance Pack in a multi-core system, after you define bonds, set affinities manually. Use the -s option of the `sim affinity` command (see *Performance Pack* documentation).

Note - sim affinity commands take effect only if the Performance Pack is enabled and actually running. Performance Pack begins running when you install a policy for the first time.

For optimal performance, set affinities according to the following guidelines:

1. Run `sim affinity` using the *-s* option.
2. Whenever possible, dedicate one processing core to each interface. See sk33520 (http://supportcontent.checkpoint.com/solutions?id=sk33250).
3. If there are more interfaces than cores, one or more cores handle two interfaces. Use interface pairs of the same position with internal and external bonds.
   a) To view interface positions in a bond, run:
      `cat /proc/net/bonding/<bond name>.`
   b) Note the sequence of the interfaces in the output, and compare this for the two bonds (external bond and its respective internal bond). Interfaces that appear in the same position in the two bonds are interface pairs and set to be handled by one processing core.

   For example, you might have four processing cores (0-3) and six interfaces (0-5), distributed among two bonds:

bond0	bond1
eth0	eth3
eth1	eth4
eth2	eth5

   Two of the cores will need to handle two interfaces each. An optimal configuration might be:

bond0		bond1	
eth0	core 0	eth3	core 0
eth1	core 1	eth4	core 1
eth2	core 2		
		eth5	core 3

Verifying that the Bond is Functioning Properly

After installation or failover, it is recommended to verify that the bond is up, by displaying bond information.

1. Run:
   ```
   cphaprob -a if
   ```
   Make sure that the bond status is reported as `UP`.
2. Run:
   ```
   cphaconf show_bond <bond name>
   ```
   Check that the bond is correctly configured.

Reconfiguring Topology

At this point, you need to reconfigure the relevant objects to connect to the newly created bond. This includes Virtual Systems, Virtual Routers and Virtual Switches. You can perform these actions using SmartDashboard. In most cases, these definitions can be found in the object **Properties** window.

For large existing VSX deployments containing many Domain Management Servers and virtual devices, use the *vsx_util change_interfaces* command to reconfigure existing object topologies. For example, in a Multi-Domain Security Management deployment with 200 Domains, each with many virtual devices, it is much faster to use *vsx_util change_interfaces*. This command automatically replaces the interface with the new bond on all relevant objects.

Reconfiguring Topology using vsx_util change_interfaces

To reconfigure objects using vsx_util change_interfaces:

Important - All Domain Management Servers must be unlocked in order for this operation to succeed.

1. Close SmartDashboard for all Multi-Domain Security Management Domain Management Servers using the designated interface.
2. On the management server, enter the Expert Mode and execute the *vsx_util change_interfaces* command.
3. Enter the Security Management Server, or Multi-Domain Security Management main Domain Management Server IP address.
4. Enter the administrator name and password as requested.
5. Enter the VSX cluster object name.
6. Select **Apply changes to the management database and to the VSX Gateway/Cluster members immediately**.
7. When prompted, select the interface to be replaced.
8. When prompted, select the replacement bond interface.
9. If you wish to replace additional interfaces, enter "**y**" when prompted and repeat steps 6 and 7.
10. To complete the process, enter "**n**".

Reconfiguring the Bond using SmartDashboard

To configure the newly created bond:

1. In the SmartDashboard navigation tree, double-click the VSX gateway or cluster object.
2. In the **Properties** window, select the **Physical Interfaces** branch.
3. Click **Add** to add the new Bond to the cluster object.
   a) In the **Physical Interface Properties** window, enter the bond name. This name must be exactly the same as the name assigned to the bond when it was created with the *sysconfig*.
   b) If the bond is a VLAN trunk, enable the **VLAN Trunk** option.
4. On the **Topology** page:
   a) Double-click the interface(s) to be replaced.
   b) Select the bond interface from the **Interface** list.
   c) Click OK to push the configuration.

5. Install the policy.

Configuring Cisco Switches for Load Sharing

These are sample configuration commands for Cisco switches.

For 802.3ad:

```
Switch#conf t
Switch(config)#port-channel load-balance src-dst-ip
Switch(config)#interface FastEthernet <all the
participating interfaces>
Switch(config-if)#channel-group 1 mode active
Switch(config-if)#channel-protocol lacp
Switch(config-if)#exit
Switch(config)#interface port-channel 1
Switch(config-if)#switchport access vlan <the wanted vlan
number>
Switch(config-if)#end
Switch#write
```

For XOR:

```
Switch#conf t
Switch(config)#port-channel load-balance src-dst-ip
Switch(config)#interface FastEthernet <all the
participating interfaces>
Switch(config-if)#channel-group 1 mode on
Switch(config-if)#exit
Switch (config)#interface port-channel 1
Switch(config-if)#switchport access vlan <the wanted vlan
number>
Switch(config-if)#end
Switch#write
```

Changing the Bond Interface Mode

You can change the bond interface mode between High Availability and Load Sharing as required for your deployment. This process of changing between bond interface modes automatically reboots the VSX Gateway or cluster members.

To change the bond interface mode:

From the VSX Gateway or cluster member command line, run sysconfig.

1. Select **Network Connections**.
2. Select **Configure Connection**.
3. Select the bond interface.
4. When prompted, confirm that you wish to continue.
5. Select a bond mode.
6. Repeat this procedure for each cluster member as required.

Enslaving Interfaces to a Bond

You can only enslave interfaces that are attached to vrf context 0.

To enslave new interfaces to an existing bond:

1. At the VSX Gateway or cluster member, run `sysconfig`.
2. Select `Network Connections`.
3. Select `Configure Connection`.
4. Select the bond interface.
5. Select `Enslave interface to bond`.
6. Select the interface to be included in the bond by entering the number corresponding to the interface name.

```
Choose interfaces to be enslaved under the bond ('n' for
next, 'e' to exit):
------------------------------------------------------
---------
1) [_]eth0  3) [_]eth2  5) [_]eth4  7) [_]eth6
2) [_]eth1  4) [_]eth3  6) [_]eth5
------------------------------------------------------
---------
(Note: configuration changes are automatically saved)
Your choice:
```

7. Repeat these steps for each cluster member.

Detaching Interfaces from a Bond

To remove interfaces from an existing bond:

1. At the VSX Gateway or cluster member, run `sysconfig`.
2. Select `Network Connections`.
3. Select `Configure Connection`.
4. Select the bond interface.
5. Select `Detach interface from bond`.
6. Select the interface by entering the number corresponding to the interface name.

```
Choose interfaces to be enslaved under the bond ('n' for
next, 'e' to exit):
------------------------------------------------------
---------
1) [_]eth0  3) [_]eth2  5) [_]eth4  7) [_]eth6
2) [_]eth1  4) [_]eth3  6) [_]eth5
------------------------------------------------------
---------
(Note: configuration changes are automatically saved)
Your choice:
```

7. Repeat this step for each cluster member.

Note - You cannot remove the only interface from a bond. If you attempt to remove an interface from a bond containing only two interfaces, a confirmation prompt appears.

Deleting a Bond

The process of deleting a bond and reconfiguring interfaces as regular interfaces or VLAN trunks consists of the following procedures:

Removing a Bond Interface from Virtual devices

You must remove the bond from all virtual devices that connect to it (Virtual Systems, Virtual Routers, Virtual Switches). You can use vsx_util show_interfaces ("show_interfaces" on page 202) to display virtual devices connected to a bond interfaces.

To remove a bond from a Virtual System:

1. In SmartDashboard, double-click the desired virtual device.
2. On the **Topology** page, select the bond and then click **Remove**.
3. Repeat this process for each virtual device connected to the bond interface.

Removing a Bond Interface From a VSX Object

You can remove a bond interface from a VSX object only when there are no virtual devices connected the bond.

To remove a bond from a VSX Gateway or cluster object:

1. In SmartDashboard, double-click the VSX Gateway or cluster object.
2. On the **Physical Interfaces** page, select the bond interface and click **Remove**.

Removing a Bond Interface from a VSX Gateway or Cluster Member

The next step is to use the *sysconfig* utility to remove a bond interface from the VSX Gateway or cluster members.

To remove a bond Interface:

1. From the VSX Gateway or cluster member command line, run *sysconfig*.
2. Select **Network Connections**.
3. Select **Remove Connection**.
4. Select the bond interface.
5. Repeat this procedure for each cluster member as required.

Reconfiguring Interface Connections

The final step in the process is to reconfigure the VSX gateway or cluster and its virtual devices definitions to connect to the former slave interfaces or VLANs.

To Reconfigure the VSX Gateway or cluster interface connection:

1. In SmartDashboard, double-click the VSX Gateway or cluster object.
2. On the **Physical Interfaces** page, click **Add**.
3. Enter the interface name in the designated field.
4. If you wish to enable this interface as a VLAN trunk, enable the option.

To reconfigure interface connections on virtual devices:

1. In SmartDashboard, double-click the appropriate virtual device.
2. On the **Topology** page, click **Add**.
3. In the **Interface Properties** window:
   a) Select an interface from the list.
   b) If the interface is defined as a VLAN trunk, enter a VLAN tag in the designated field.
   c) Enter the interface IP address and net mask.
   d) Optionally enable propagation to adjacent virtual devices.

Changing an Existing Interface to a Bond

The following sample scenario demonstrates the procedure for configuring an existing VSX cluster to a use a Link Aggregation bond. The VSX cluster members currently uses interface eth1 to connect to several Virtual Machines and other virtual devices. Interface eth 2 is currently free and eth0 serves as the management interface.

To create a new bond using eth1 and eth2 as slave interfaces:

1. On each member, create a new bond0 ("Defining the Interface Bond" on page 134) using only eth2 as the slave interface.
2. On the management computer, use the vsx_util change_interfaces ("change_interfaces" on page 197) command to replace eth1 with the new bond0.
3. Enslave eth1 to bond0 ("Enslaving Interfaces to a Bond" on page 145) on each member.
4. In SmartDashboard, remove eth1 and eth 2 from the VSX gateway ("Removing IP Addresses from Slave Interfaces" on page 135) physical interfaces.

Troubleshooting Bonded Interfaces

Troubleshooting Workflow

1. Check the status of the bond, as detailed in ("Verifying that the Bond is Functioning Properly" on page 135).
2. If there is a problem, check if the physical link is down, as follows:
   a) Execute the following command:
      `cphaconf show_bond <bond-name>`
   b) Look for a slave interface that reports the status of the `link` as *no*.
   c) Check the cable connections and other hardware.
   d) Check the port configuration on the switch.
3. Check if a cluster member is down, by running:
   `cphaprob state`
   If any of the cluster members have a `Firewall State` other than *active*, see "*Monitoring Cluster Status (cphaprob state)*" in the *R75.20 ClusterXL Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12265)for further troubleshooting help.
4. For further information regarding bond status and failovers, view logs in SmartView Tracker. Any interface bond status change is logged and can be viewed in SmartView Tracker

Connectivity Delays on Switches

When using certain switches, connectivity delays may occur during some internal bond failovers. With the various features that are now included on some switches, it can take close to a minute for a switch to begin servicing a newly connected interface. The following are suggestions for reducing the startup time after link failure.

1. Disable auto-negotiation on the relevant interface.
2. On some Cisco switches, enable PortFast, as detailed below.

Note - PortFast is not applicable if the bond group on the switch is configured as Trunk.

Warning Regarding Use of PortFast

The PortFast feature should never be used on ports that connect to other switches or hubs. It is important that the Spanning Tree complete the initialization procedure in these situations. Otherwise, these

connections may cause physical loops where packets are continuously forwarded (or even multiply) in such a way that network will ultimately crash.

Sample Configuration of PortFast on a Cisco Switch

The following are the commands necessary to enable PortFast on a Gigabit Ethernet 1/0/15 interface of a Cisco 3750 switch running IOS.

1. Enter configuration mode:
   ```
   cisco-3750A#conf t
   ```
2. Specify the interface to configure:
   ```
   cisco-3750A(config)#interface gigabitethernet1/0/15
   ```
3. Set PortFast on this interface:
   ```
   cisco-3750A(config-if)#spanning-tree portfast
   ```

Chapter 9

Optimizing VSX

In This Chapter

VSX Resource Control 150
QoS Enforcement 153

VSX Resource Control

Overview

VSX Resource Control allows administrators to ensure that critical traffic receives a greater share of the available VSX gateway or cluster member processing power by assigning priorities to each Virtual System. Since individual networks protected by Virtual Systems have differing traffic loads, user demand, and criticality, allocating available CPU resources to individual Virtual Systems according to predetermined priorities is an important consideration in any VSX deployment.

By using VSX Resource Control, administrators can allocate CPU resources to individual Virtual Systems according to specific requirements. Virtual Systems protecting mission critical networks or resource intensive applications are assigned higher priorities and are ensured a greater share of CPU resources. Virtual Systems with lower priorities are guaranteed a lesser CPU share.

In addition, VSX Resource Control performs the following actions:

- Ensures that each Virtual System receives a minimum CPU allocation and that no Virtual System can deny another Virtual System access to the CPU
- Makes CPU resources not required by Virtual Systems to which they are allocated available for use by other Virtual Systems, until they are needed

VSX Resource Control can only manage **available CPU resources**, meaning those CPU resources that are not required by other processes running on the VSX gateway. If the operating system, Check Point products and other application processes consume 40% of CPU resources, VSX Resource Control can only manage the remaining 60%.

Resource Control System Components

Resource Control has two principal components:

- The **Resource Control Monitor** keeps track of the CPU consumption of each Virtual System. It also provides real-time information on the present and average CPU consumption by the Virtual Systems on the VSX machine.
- The **Resource Control Enforcer** implements the Resource Policy. The Resource Control Enforcer utilizes the data collected by the Resource Control Monitor to implement the Resource Policy.

Note - You must enable the Resource Control Monitor in order to use the Resource Control Enforcer. You can, however, activate the monitor without the enforcer if only wish to monitor CPU utilization.

Virtual System Priorities

VSX Resource Control uses a **weight** factor to assign priorities to Virtual Systems. The weight factor is expressed as an integer between one and 100, that indicates a particular Virtual System's priority in relation to other Virtual Systems. Virtual Routers and Virtual Switches are automatically assigned to priority and their priorities are not modifiable.

The Default Weight Factor

Each new or undefined Virtual System receives a default weight to 10 by default. This means that they are entitled to an equal proportion of available CPU Resources. For example, on a VSX gateway containing five Virtual Systems with the default weight factor of 10, each is guaranteed at least 20% of the available CPU resources. Adding five more Virtual Systems with default weights to this same gateway would guarantee each Virtual System a minimum of 10% of the available resources.

User Defined Weights

To assign different priorities to individual Virtual Systems, you must manually assign a weight factor to each. The percentage of CPU resources guaranteed to a particular Virtual System is determined by dividing its weight by the total weight of all Virtual Systems on the VSX Gateway or cluster.

For example, if Virtual System A is assigned a weight of 40, and the total weight of all Virtual Systems (including Virtual System A) is 80, then Virtual System A is guaranteed a minimum of 50% of the available CPU resources.

Note - As long as the VSX gateway has available CPU resources, each Virtual System receives some resources regardless of its weight. When a system is under stress, however, Virtual Systems with the lower weights will be the first to be affected by the lack of CPU resources.

Working with VSX Resource Control

This section presents procedures for enabling and disabling VSX Resource Control, for assigning Virtual System priorities and for monitoring CPU resource utilization. VSX Resource Control is disabled by default.

Enabling VSX Resource Control Components

To enable VSX Resource Control enforcement, execute the `vsx resctrl enforce enable` command. This command starts both the **Resource Control Enforcer** and the **Resource Control Monitor**.

To enable the Resource Control Monitor without enforcement, execute the `vsx resctrl monitor enable` command.

Disabling VSX Resource Control Components

To enable VSX Resource Control enforcement, execute the `vsx resctrl enforce` command. This command starts both the **Resource Control Enforcer** and the **Resource Control Monitor**.

To disable the Resource Control Enforcer, execute the `vsx resctrl enforce disable` command.

To disable the Resource Control Monitor, execute the `vsx resctrl monitor disable` command.

You execute each command separately.

Note - The `vsx resctrl enforce` and `vsx resctrl monitor` commands work for the current session only. Upon reboot, Resource Control is automatically enabled or disabled according to settings in the Resource Control configuration file.

Assigning Priorities to Virtual Systems

Assigning Virtual System priorities requires editing the Resource Control configuration file `$FWDIR/conf/resctrl` on the VSX gateway or on each cluster member. The following information appears in the configuration file:

- Resource Control Monitor default setting (*enabled/disabled*). This if
- Resource Control Enforcer default setting (*enabled/disabled*)
- Manually assigned Virtual System weight factors. Virtual Systems using the default weight factor (10) do not appear in the configuration file.

A sample configuration file appears below:

```
#file format:
#first line must be "monitoring enabled" or "monitoring disabled"
#second line must be "enforcement enabled" or "enforcement
disabled"
#rest of the lines should contain weights for the virtual systems
where
#weight varies between 1 and 100
#<vsid> <weight>
#For example:
#8 40

monitoring enabled
enforcement enabled
2 25
3 40
4 15
```

To manually assign Virtual System priorities:

1. Open the `$FWDIR/conf/resctrl` file using a text editor.
2. Change the **monitoring** and **enforcement** default setting to **enabled**. This ensures that the Resource Control enforcer is active by default whenever the VSX Gateway or cluster member is rebooted.
3. For each Virtual System, enter its Virtual System ID (`vsid`) followed by a space and its weight factor (an integer between 1 and 100 inclusive). Each Virtual System must appear on a separate line, as illustrated in the above example.
4. Save and close the configuration file.
5. Execute the `vsx resctrl start` command.
6. Execute the `vsx resctrl stat` command to verify the configuration.

Monitoring CPU Resource Utilization

You can view real time statistical information about CPU resource utilization. To do so, execute the `vsx resctrl stat` command. A screen similar to the following appears.

```
[Expert@rescon:0]# fw vsx resctrl stat

Virtual Systems CPU Usage Statistics
====================================

Number of CPUs/Hyper-threading: 4
Monitoring active time: 14s

ID    Name               |Weight|  1sec  10sec   1min    1hr   24hr*
=======================+======+=================================
  0   VSX2               |  N/A |  0.11   0.06   0.08   0.07   0.01
  1   VSX2_vs1           |  10  | 15.80  21.57  21.75  22.28   1.94
  2   VSX2_vsw           |  N/A |  0.00   0.00   0.00   0.00   0.00
  3   VSX2_vs2           |  10  | 16.91  22.57  22.77  23.09   2.01
=======================+======+=================================
Total VS CPU Usage             | 32.82  44.20  44.60  45.44   3.96
===============================+=================================
System CPU Usage               |    99     99     99     99     89

Notes:  - Monitoring has been active for less than 24 hours.
          Statistics are calculated only for monitoring active time.
        - The displayed statistics are the average usage on all
CPUs.
```

Notes

- For systems with more than one CPU, the time displayed is an average among all CPUs. To view usage for each Virtual System per CPU, run the command `vsx resctrl -u stat`.
- The VSX gateway itself, Virtual Switches and Virtual Routers are not assigned weights because they always receive the highest priority
- **Total VS CPU Usage** represents the total CPU utilization for all virtual devices, including Virtual Routers, Virtual Switches and the VSX gateway
- **System CPU Usage** reports the total CPU utilization for the entire machine

QoS Enforcement

Overview

QoS Enforcement for VSX provides the ability to control the network quality of service in the VSX network environment. QoS is based on the Differentiated Services architecture and allows assigning different transmission characteristics to different classes of service.

Differentiated Services is a computer networking architecture that specifies a simple, scalable and coarse-grained mechanism for classifying, managing network traffic and providing quality of service (QoS) guarantees on modern IP networks. Differential services can, for example, be used to provide low-latency, guaranteed service (GS) to critical network traffic such as voice or video while providing simple best-effort traffic guarantees to non-critical services such as web traffic or file transfers.

The major characteristics that are controllable by QoS are latency and bandwidth allocation. QoS is designed to provide QoS functionality with minimal impact on performance. QoS works seamlessly with Check Point Performance Pack.

The VSX network usually includes various types of traffic such as:

- Real-time traffic (e.g. VoIP) which requires low bandwidth, and is sensitive to latency (delays) and drops
- Traffic which is sensitive to latency but not to occasional drops
- High-volume, low-priority traffic which has a low sensitivity to latency and drops
- Other traffic which requires its own share of the bandwidth

Without QoS Enforcement, all these different traffic types are given equal priority on the VSX gateway and are handled in a simple FIFO (first in-first out) manner. When the VSX gateway is congested, all traffic types suffer the same degree of latency and drops. Also, high-volume traffic may starve other types of low-volume traffic.

With QoS, the special requirements of each traffic type can be met. For example:

- Latency-sensitive traffic will be given preference over other types of traffic
- Traffic which is sensitive to drops will suffer fewer drops than other types of traffic.
- High-volume traffic that consumes bandwidth will be limited during times of congestion.

Note - QoS requires the use of DiffServ-enabled routers to mark preferred traffic types with a special tag. The tag is the DSCP (DiffServ Code Point), which represents the six most significant bits of the IP header's TOS field, as described in RFC 2474. The VSX gateway should then be configured to give traffic with this tag the required priority.

Architecture

Three major aspects of the QoS architecture are:

- Differentiated Services support
- Inbound prioritization
- Policy with a global scope

Differentiated Services Support

QoS provides basic support for Differentiated Services, an architecture for specifying and controlling network traffic by class so that certain types of traffic receive priority over others. The differentiated services architecture PHB's (per-hop behaviors).

When marked packets arrive to the VSX machine, they are classified and prioritized according to their DSCP (differential services code-point) values. To enhance performance, QoS does not mark packets with DSCP and does not change their Type of Service (ToS) values. QoS instead relies on peripheral devices (namely routers) to mark packets with the appropriate ToS value.

Inbound Prioritization

While Differentiated Services support in routers is usually performed on outbound traffic, QoS for VSX prioritizes traffic on the inbound side because, in VSX deployments, QoS is primarily governed by system resources, namely the CPU, and not by network bandwidth.

To prevent the VSX machine from becoming a bottleneck in the network, prioritization is enforced when packets arrive at the VSX machine, and before CPU processing is assigned.

Inbound prioritization allows an earlier control on the loss and delay rate.

Policy with Global Scope

To minimize the impact of QoS functionality on performance, QoS is not performed on a per interface basis, but for the entire system. This means that a certain class of service will apply to all traffic entering the VSX gateway or cluster, regardless of the specific interface from which the traffic originates.

Note - On multiple-CPU machines, enforcement is not performed system-wide, but executed per-CPU. This means that global enforcement is done separately on traffic processed by each CPU.

QoS Features

Two main features of QoS are:

- Resource allocation
- Latency control

Resource Allocation

System resources are allocated by assigning different weights to different classes of service. A weight is the relative portion of the available resources allocated to a class. Allocating resources according to weights ensures full utilization of the line even if a specific class is not using all of its resources. In such a case, the remaining resources are divided among the remaining classes in accordance with their relative weights.

Latency

For some types of traffic, such as voice and video, it is necessary to minimize the latency (delay) of packets. Latency is controlled by defining special **LLQ** (low-latency queueing) classes. These classes are handled in a strict priority manner. LLQ packets are handled immediately upon arrival, and before packets that do not belong to LLQ classes.

QoS supports multiple LLQ classes. In some cases, it may be necessary to define more than one Low Latency class, for example when different types of traffic have a different sensitivity to delays. In such cases, a class with the higher sensitivity to delay receives a higher priority than a class with the lower sensitivity.

Note - When LLQ classes are used, it is assumed that the expected traffic will not exceed a relatively small amount of the available resources. Although QoS does not allow LLQ traffic to starve non-LLQ traffic, too much LLQ traffic reduces overall network quality of service and prevents efficient management of weighted resources.

WRED

RED (Random Early Drop) is a congestion avoidance mechanism for detecting and preventing congestions. It takes advantage of TCP's congestion control mechanism by randomly dropping packets during periods of congestion. This causes TCP senders to slow down their transmission, thus preventing high congestion.

QoS implements WRED (Weighted RED) in which packets are dropped according to their priority. WRED mostly affects traffic which is of low priority and which exceeds its weight.

QoS Management

To manage the network quality of service it is necessary to create and install a QoS policy. The QoS policy consists of a list of up to 15 classes of service. Each class is assigned certain traffic characteristics and DSCP values.

The QoS policy is managed using the cpqos ("The cpqos Command" on page 156) command.

Class of Service Definitions

The definition of a class of service includes the following:

- **Name**. The class name is a unique identifier which identifies the class during configuration and when presenting statistics
- Type. There are two types of classes, LLQ and regular classes. Regular classes are non-LLQ classes which can be assigned a weight value.
- Priority. Each class is assigned a unique priority value between 1 and 15. The priority value is effective in prioritizing LLQ classes and during congestion, when drops occur.
- Weight value. Each class is assigned a specific weight value

- One or more DSCP values. The Differentiated Services code point

Priority and LLQs

If there are multiple LLQ classes, packets are handled in a strict priority-based manner. Packets from a class with a higher priority are handled before packets with a lower priority class.

Priority and Drop Precedence

Priority also determines the probability of drops. A class with a lower priority has a higher drop precedence during times of congestion.

The class priority is not the only factor that determines if drops occur. Other factors affect drops, for example if the class is LLQ or if the class exceeds its assigned resource allocation.

LLQ's are not immune to drops. Although LLQ's are processed as soon as they arrive (and thus have a lower drop rate), drops may occur if there are many LLQ classes or if a large portion of the incoming traffic is LLQ.

QoS Configuration

All user interactions with the QoS module are performed with the `cpqos` command.

The cpqos Command

```
cpqos - Manage the network quality of service.

cpqos install            - Install the QoS policy.
cpqos uninstall          - Uninstall the QoS policy.
cpqos status             - Check if policy is installed.
cpqos class show [-b]    - Show the QoS policy. [-b] display dscp in
binary numbers.
cpqos class add <name> prio <val> type <llq|reg> [weight <val>] dscp
<val[,val2[,val3...]]>
                         - Add new class with specified name
                           prio   - priority (1-15)
                           type   - low-latency or regular
                           DSCP   - values "default" or 0-63
(decimal or 8 bits binary)
                           weight - (1-1000) for classes of type
"reg"
cpqos class del <name>   - Delete specified class name.
cpqos stats [-u]         - Show statistics. [-u] will show
statistics per CPU.
```

For cpqos:

- All commands return a zero value for success and a non-zero value for failure
- Options and argument are case-sensitive
- Examples of various `cpqos commands` ("Sample Differentiated Services Implementation" on page 158)

cpqos class add

This command adds a class with the following arguments:

Argument	Value
name	Unique name for the class
priority	Value between 1 and 15. A low value indicates a higher priority
type	"llq" for low-latency classes or "reg" for regular, weighted classes

Argument	Value
weight	This value is used only for classes of type "reg". It determines the relative portion of the resources that the class will receive in relation to other weighted classes. Valid values are between 0 and 1000.
dscp	The DiffServ code-points assigned to the class. Multiple DSCP's can be specified, separated by commas, with no spaces between values. Values are in decimal (not binary format) with values from 0 to 63 or "default". There can be only one class with a "default" DSCP. The default class is used for traffic without DiffServ marking (e.g. tos=0) or traffic with DSCP values that are not assigned to any other class. If no class is used as "default", all 64 DSCP values must be assigned to the classes. A DSCP value cannot be assigned to more than one class.

Note - Changes to the policy with cpqos class add are enforced only after the policy is installed.

cpqos class del

This command deletes the class of the specified name. Changes to the policy with `cpqos class del` are enforced only after the policy is installed.

cpqos class show

This command shows the classes defined in the QoS policy.

cpqos install

This command installs the previously created QoS policy. It also validates the overall integrity of the policy. Once installed, the policy remains installed even if the machine reboots.

cpqos uninstall

This command un-installs the previously installed QoS policy. If un-installed, the policy will not be installed again when the machine boots.

cpqos stats

This command shows QoS statistics. `cpqos stats` prints a line of statistics for each of the defined classes. Each line includes the following data columns:

Column	Value
rx	Number of bytes that arrived to this class since the last time statistics were presented
tx	Number of bytes that were transmitted from this class since the last time statistics were presented
drops	Number of bytes that were dropped from this class since the last time statistics were presented

Note:

- [-u] option shows statistics separately for each CPU
- Statistics values are byte-counts, not packet counts

- Statistics values are reset after each query.
- Statistics should be presented periodically with intervals less than 1 minute.
- It is recommended to use the `watch` command to periodically present the statistics.

QoS Policy File

The QoS policy file is `qos_policy.C`, located in the `$FWDIR/database` directory. The QoS policy file is created when the `cpqos` command is run for the first time. The QoS policy file should not be edited manually. Use `cpqos class add/del` to create entries. To maintain multiple QoS policies, rename `qos_policy.C` or copy it to another directory, and copy it back to `$FWDIR/database/qos_policy.C` when the policy needs to be enforced.

QoS Default Configuration

Default QoS configuration is set to "uninstall" (e.g. not enforced). Calling `cpqos install` or `cpqos uninstall` sets the default configuration after boot

Sample Differentiated Services Implementation

This section presents a sample differentiated services implementation. It includes examples for configuration, monitoring and statistics.

Sample Traffic Types

Traffic Type	Meaning...
Diamond	Real-time traffic (e.g. VOIP) which requires little bandwidth and is sensitive to latency and drops. This traffic is usually assigned to the EF (Expedited-Forwarding) PHB (Per Hop Behavior).
Platinum	Real-time traffic with low bandwidth requirements that is less sensitive to latency and drops than Diamond.
Gold	Traffic which is sensitive to drops
Silver	Traffic which is less sensitive to drops than Gold.
Bronze	Various types of traffic which require resource allocation. This traffic is usually assigned to the Best-Effort PHB.
Copper	High-volume traffic with a tendency to consume bandwidth

Configuration Guidelines

Your QoS policy should apply these guidelines:

- Diamond and Platinum classes should be defined as LLQ so they will have a lower latency then other classes
- Diamond should receive a higher priority than Platinum so it have even less latency and drops
- Gold should receive a higher priority than Silver so it will have fewer drops
- Copper resource consumption should be limited to about 10% of the available resource during periods of congestion
- Other traffic should receive about 45% of bandwidth when the traffic load is high

Configuration Examples

1. The following examples of the `cpqos class add` command creates classes for traffic of various types:

```
cpqos class add Diamond type llq prio 1 dscp 46
```

```
cpqos class add Platinum type llq prio 2 dscp 32
cpqos class add Gold type reg prio 3 weight 100 dscp 26
cpqos class add Silver type reg prio 4 weight 100 dscp 28
cpqos class add Bronze type reg prio 5 weight 200 dscp
default
cpqos class add Copper type reg prio 15 weight 50 dscp
10,12,14
```

2. Monitoring example. The following command lists the previously defined classes:

```
[Expert@cpmodule:0]# cpqos class show
class: Diamond
      priority: 1
      type: llq
      weight: 0
      DSCPs: 46

class: Platinum
      priority: 2
      type: llq
      weight: 0
      DSCPs: 32

class: Gold
      priority: 3
      type: reg
      weight: 100
      DSCPs: 26

class: Silver
      priority: 4
      type: llq
      weight: 100
      DSCPs: 28

class: Bronze
      priority: 5
      type: llq
      weight: 200
      DSCPs: default

class: Copper
        priority: 15
        type: reg
        weight: 50
        DSCPs: 10,12,14
```

3. Statistics example. The following command lists statistics for the previously defined classes:

```
class     priority type       weight     rx         tx
     drops
Diamond   1        llq        0          2775       2650
     0
Platinum  2        llq        0          1024       1020
     105
Gold      3        reg        100        1775015    17738
05   205
Silver    4        reg        100        1862437    18623
36   550
Bronze    5        reg        200        3370033    29551
20   3147
Copper    15       reg        50         1862437    76233
6    100689
```

From this statistical output, it is apparent that:

- In the Diamond class there were no drops.
- In the Platinum class there were a few drops, even though less traffic arrived classed as Platinum than did as Diamond.
- In the Gold class there were fewer drops than from the Silver class.
- In the Bronze class there were twice as many bytes transmitted than in the Silver and Gold classes, and four times as many bytes than there were in the Copper class.
- Most packets in the Copper class were dropped.

Chapter 10

Hardware Health Monitoring

SecurePlatform enables a number of hardware health monitoring capabilities for Check Point appliances and for open servers.

In This Chapter

Introduction to Hardware Health Monitoring	161
RAID Monitoring with SNMP	162
Sensors Monitoring with SNMP on VSX-1 Appliances	163

Introduction to Hardware Health Monitoring

SecurePlatform features the following Hardware Health Monitoring capabilities:

- **RAID health:** Monitor the health of the disks in the RAID array, and be notified of the states of the volumes and disks. The information is available via SNMP.
- **Sensors:** Monitor fan speed, motherboard voltages and temperatures on the hardware. The information is available via SNMP and, for Check Point appliances.

The following matrix summarizes the supported health monitoring features:

	Check Point Appliances		Open Servers			
	VSX -1 9070 11070	VSX -1 3070	IBM	HP	Dell	Sun
Hardware sensors monitoring with SNMP (polling and traps)	X	X[1]	X[2]	X[2]	X[2]	X[2]
Hardware sensors monitoring with Web UI (Polling and Traps)	X	X[1]				
RAID Monitoring with SNMP	X[3]			X[4]		

1. **Hardware sensors monitoring** is supported on all VSX-1 models.
2. **Hardware sensors monitoring** for open servers is supported on certified servers with an Intelligent Platform Management Interface (IPMI) card installed. The IPMI specification defines a set of common interfaces to a computer system, which system administrators can use to monitor system health.
3. **RAID Monitoring with SNMP** is supported on VSX-1 servers with a RAID card installed (VSX-1 9070 and VSX-1 11070).
4. **RAID Monitoring with SNMP** on HP servers is supported with a P400 RAID controller.

Note - Hardware Monitoring is enabled by default on all VSX-1 appliances and disabled by default on all Open Server platforms.

To enable Hardware Monitoring on an Open server, select the **Hardware Monitoring** option in `Sysconfig`.

RAID Monitoring with SNMP

The health of disks' RAID array can be monitored using the SecurePlatform SNMP monitoring daemon. SNMP traps can be set to fire once an OID value is in breach of a configurable threshold.

The raidInfo MIB branch is 1.3.6.1.4.1.2620.1.6.7.7. The information it contains is detailed below.

Data is available in the form of two SNMP tables:

SNMP Table	OID
Volumes	1.3.6.1.4.1.2620.1.6.7.7.1.1
Disks	1.3.6.1.4.1.2620.1.6.7.7.2.1

Each volume in the RAID configuration has an entry in the Volumes table. Each volume's entry in the Volumes table contains the following OID values:

Disk Volume Information	OID	Comment
Index	.1	
Volume ID	.2	
Volume Type (RAID level)	.3	For check Point appliances, will normally be RAID_1
Number Of Disks in the RAID	.4	
Volume size	.5	Maximum supported LBA (Logical Block Addressing)
Volume state	.6	One of: • OPTIMAL • DEGRADED • FAILED
Volume state	.7	One or more of: • ENABLED • QUIESCED • RESYNC_IN_PROGRESS • VOLUME_INACTIVE

Each disk participating in the RAID configuration has an entry in the disks table. Each disk's entry in the table contains the following OID values:

Physical Disks information	OID	Comment
Index	.1	
Volume ID	.2	
SCSI ID	.3	
Disk number	.4	On Check Point Power-1 9070 appliance: 0 - upper disk, 1 - lower disk
Vendor	.5	
Product ID	.6	

Physical Disks information	OID	Comment
Revision	.7	
Size	.8	Maximum supported LBA (Logical Block Addressing)
State	.9	One of the following: • ONLINE • MISSING • NOT_COMPATIBLE • FAILED • INITIALIZING • OFFLINE_REQUESTED • FAILED_REQUESTED • OTHER_OFFLINE
Flags	.10	One of: • OUT_OF_SYNC • QUIESCED
Sync state	.11	A percentage. Shows how much of the backup disk is synchronized with the primary disk

Example RAID Monitoring OIDs

OID	Meaning
1.3.6.1.4.1.2620.1.6.7.7.1.1.3.1.0	RAID type field of entry #1 from the volumes table
1.3.6.1.4.1.2620.1.6.7.7.1.1.4.1.0	Number of disks field of entry #1 from the volumes table
1.3.6.1.4.1.2620.1.6.7.7.2.1.4.2.0	Disk number field of entry #2 from the disks table

SNMP monitoring rules are defined in the `snmpd.conf` configuration file. For full details see SNMP Monitoring.

Sensors Monitoring with SNMP on VSX-1 Appliances

Note - The information in this section is taken from SecureKnowledge solution sk42426 ('http://supportcontent.checkpoint.com/solutions?id=sk2426)

On Power-1 and UTM-1 appliances the hardware status can be monitored using WebUI and SNMP polling, or by defining the SNMP trap using the `cp_monitor` mechanism.

SNMP monitoring rules are defined in the `snmpd.conf` configuration file. For full details see SNMP Monitoring. Examples of `cp_monitor` for various appliance types are as follows:

VSX-1 3070

```
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.1.0 > 80 20 "M/B Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.2.0 > 100 20 "CPU Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.1.0 < 4220 20 "Case Fan speed is too low"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.2.0 < 16320 20 "CPU 1 Fan speed is too low"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.3.0 < 16320 20 "CPU 2 Fan speed is too low"
```

VSX-1 9070

```
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.1.0 > 100 20 "CPU 1 Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.2.0 > 100 20 "CPU 2 Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.3.0 > 80 20 "M/B Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.1.0 < 3000 20 "CPU 1 Fan speed is too
low"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.2.0 < 3000 20 "CPU 2 Fan speed is too
low"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.3.0 < 3000 20 "Case Fan speed is too low"
```

VSX-1 11070

```
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.1.0 > 100 20 "CPU 1 Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.2.0 > 80 20 "M/B Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.3.0 > 100 20 "CPU 2 Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.1.0 < 0 20 "Case Fan 1 speed is too low"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.2.0 < 500 20 "CPU 1 Fan speed is too low"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.3.0 < 500 20 "CPU 2 Fan speed is too low"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.4.0 < 0 20 "Case Fan 2 speed is too low"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.5.0 < 0 20 "Case Fan 3 speed is too low"
```

Example Sensors Monitoring OIDs

OID	Meaning
1.3.6.1.4.1.2620.1.6.7.8.1.1.2.1.0	Name of sensor #1 from the Temperatures table
1.3.6.1.4.1.2620.1.6.7.8.2.1.3.4.0	Value of sensor #4 from the Fan Speeds table
1.3.6.1.4.1.2620.1.6..7.8.3.1.6.5.0	Status of sensor #5 from the Voltages table

SNMP monitoring rules are defined in the `snmpd.conf` configuration file. For full details see SNMP Monitoring.

Sensors Monitoring with SNMP on Power-1 and UTM-1 Appliances

Note - The information in this section is taken from SecureKnowledge solution sk42426 ('http://supportcontent.checkpoint.com/solutions?id=sk2426)

On Power-1 and UTM-1 appliances the hardware status can be monitored using WebUI and SNMP polling, or by defining the SNMP trap using the `cp_monitor` mechanism.

SNMP monitoring rules are defined in the `snmpd.conf` configuration file. For full details see SNMP Monitoring. Examples of `cp_monitor` for various appliance types are as follows:

UTM-1 130

```
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.1.0 > 80 20 "M/B Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.2.0 > 90 20 "CPU Temp is too high"
```

UTM-1 270

```
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.1.0 > 80 20 "M/B Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.2.0 > 100 20 "CPU Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.1.0 < 16320 20 "Case Fan speed is too
low"
```

UTM-1 570 and UTM-1 1070

```
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.1.0 > 80 20 "M/B Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.2.0 > 100 20 "CPU Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.1.0 < 4220 20 "CPU Fan speed is too low"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.2.0 < 16320 20 "Case Fan speed is too
low"
```

UTM-1 2070 and UTM-1 3070

```
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.1.0 > 80 20 "M/B Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.2.0 > 100 20 "CPU Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.1.0 < 4220 20 "Case Fan speed is too low"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.2.0 < 16320 20 "CPU 1 Fan speed is too low"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.3.0 < 16320 20 "CPU 2 Fan speed is too low"
```

Power-1 5070

```
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.1.0 > 80 20 "M/B Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.2.0 > 100 20 "CPU Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.1.0 < 4220 20 "Case Fan speed is too low"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.2.0 < 16320 20 "CPU 1 Fan speed is too
low"
cp monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.3.0 < 16320 20 "CPU 2 Fan speed is too
low"
```

Power-1 9070

```
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.1.0 > 100 20 "CPU 1 Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.2.0 > 100 20 "CPU 2 Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.3.0 > 80 20 "M/B Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.1.0 < 3000 20 "CPU 1 Fan speed is too
low"
cp monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.2.0 < 3000 20 "CPU 2 Fan speed is too
low"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.3.0 < 3000 20 "Case Fan speed is too low"
```

Power-1 11000

```
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.1.0 > 100 20 "CPU 1 Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.2.0 > 80 20 "M/B Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.1.1.3.3.0 > 100 20 "CPU 2 Temp is too high"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.1.0 < 0 20 "Case Fan 1 speed is too low"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.2.0 < 500 20 "CPU 1 Fan speed is too low"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.3.0 < 500 20 "CPU 2 Fan speed is too low"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.4.0 < 0 20 "Case Fan 2 speed is too low"
cp_monitor 1.3.6.1.4.1.2620.1.6.7.8.2.1.3.5.0 < 0 20 "Case Fan 3 speed is too low"
```

Chapter 11

Deploying VSX

In This Chapter

Introduction	166
Internal Network Deployment Strategies	166
Organizational Deployment Strategies	172
Migrating from an Open Server to a VSX-1 Appliance	177

Introduction

This chapter presents deployment concepts and strategies for exploiting VSX virtualization and its unique feature set. This presentation is divided into two sections as follows:

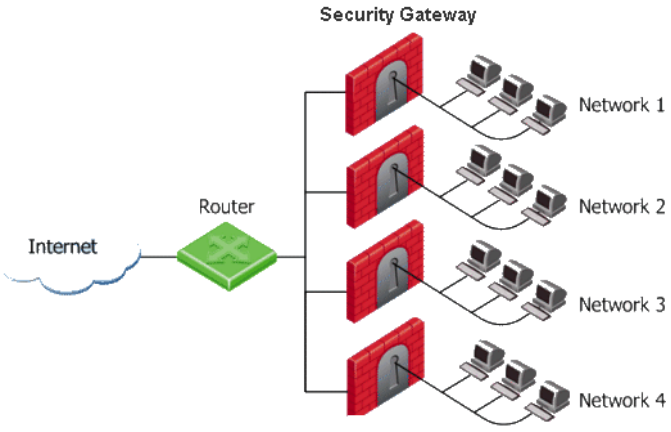
- **Internal Network Protection Strategies:** Presents concepts and examples for protecting internal network resources, including a comparison between physical deployments and VSX virtualization.
- **Organizational Deployment Strategies:** Presents VSX deployment strategies and features for several different types of large organizations.

Internal Network Deployment Strategies

Security Gateway Deployment on a Physical Network

In large physical network deployments, multiple Check Point security products, such as Security Gateways or UTM-1 Edge appliances, are deployed to protect various network segments.

Figure 11-31 Separate physical gateways protecting each network



Each Security Gateway physically connects to its own internal protected network as well as to a router for access to other internal networks and the Internet.

VSX Virtual System Deployment Strategies

In a VSX environment, Virtual Systems protect internal networks, much in the same manner as Security Gateways and other Check Point products in a physical network. This section presents several sample VSX deployments using Virtual Systems to protect internal networks.

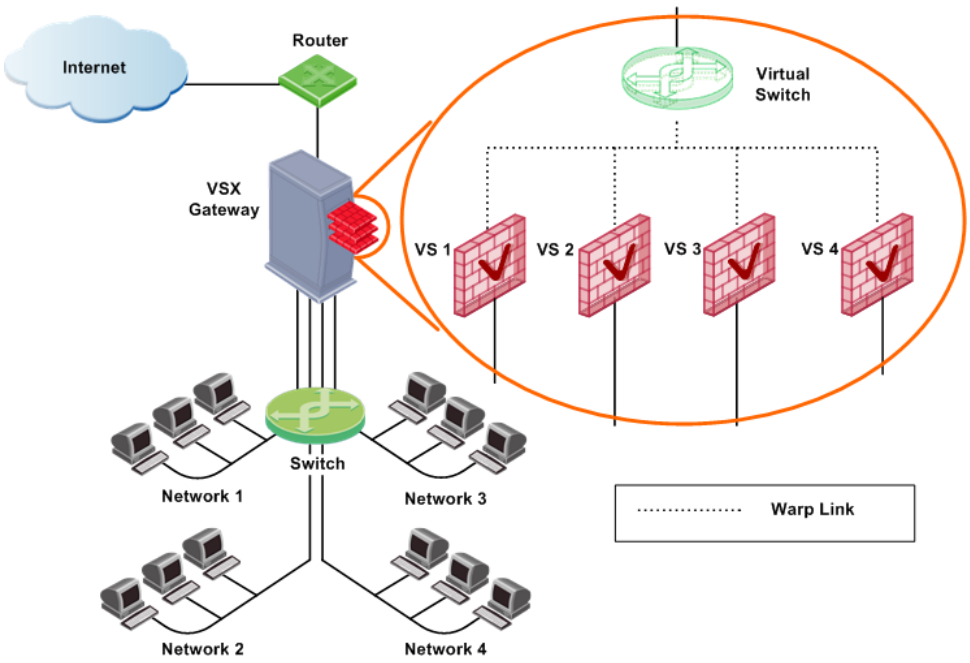
Each example highlights certain VSX features. In an real-world deployment, you can combine several of the concepts presented in these examples to create a powerful network security solution for complex enterprise environments.

Physical Internal Interface for Each Virtual System

The figure below shows a basic VSX configuration where Virtual Systems connect directly to protected internal networks using physical interfaces on the VSX gateway. A Virtual Switch provides connectivity between internal networks, as well as to the Internet. This deployment is simple to provision and is suitable for protecting a small, fixed quantity of internal networks.

The main disadvantage of this deployment is that each protected network requires its own dedicated physical interface on the VSX gateway. Obviously, this deployment is not suitable for networks that require many Virtual Systems.

Figure 11-32 Physical interface per Virtual System

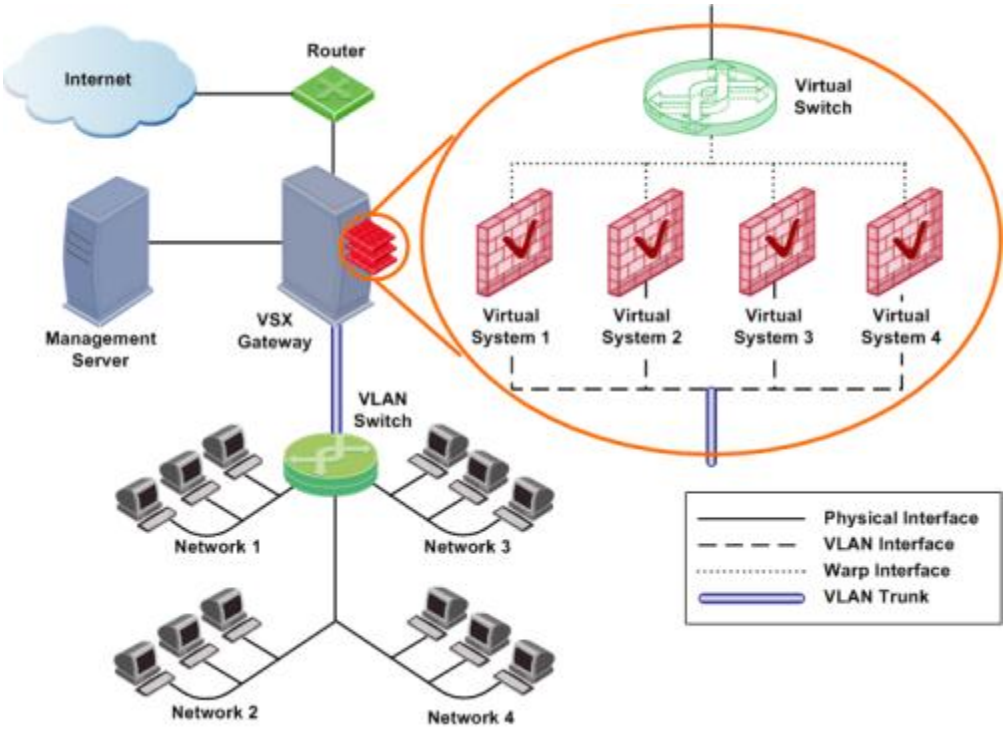


Virtual Systems with Internal VLAN Interfaces

In this deployment example, Virtual Systems connect to internal protected networks using VLAN interfaces. The VSX gateway connects to a VLAN switch via an 802.1q VLAN trunk, which is an aggregate of all VLANs passing through it.

This deployment option is appropriate for environments where many Virtual Systems protect many internal networks with a single VSX gateway or cluster. The use of VLANs provides scalability as well as granularity, allowing administrators to provision additional Virtual Systems and protected networks quickly and without impacting the existing IP address structure.

Figure 11-33 VSX interface types

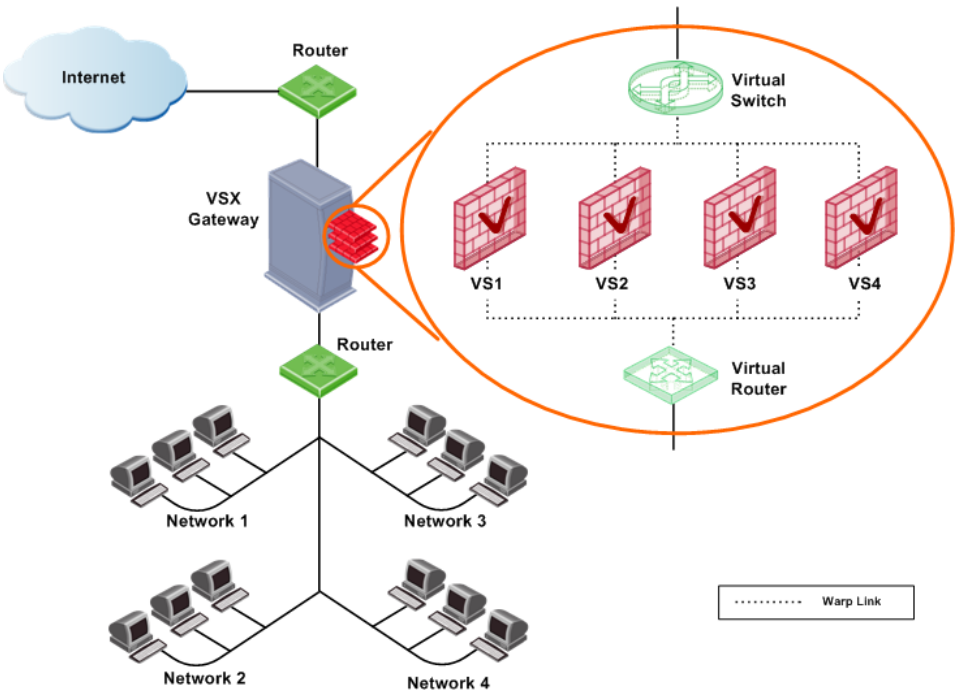


Internal Virtual Router with Source-Based Routing

This deployment scenario enables Virtual Systems to connect to protected networks using a single physical interface without VLAN technology. The Virtual Router uses source-based routing rules to forward traffic to the appropriate Virtual System based on its source IP address.

The figure below illustrates a VSX deployment with each Virtual System connected to a single Virtual Router. The Virtual Router uses source-based routing rules to forward traffic to the appropriate Virtual System based on the source IP address.

Figure 11-34 Source-based routing



Note to this scenario:

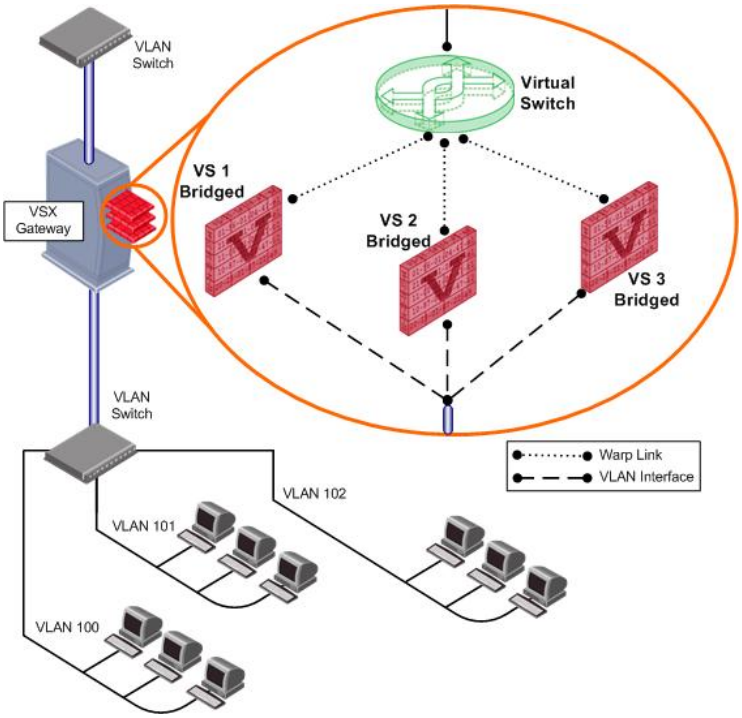
- Each Virtual System uses a public IP address to connect to the Virtual Switch
- Each local network connected to a Virtual Router uses private IP addresses
- This deployment does not support overlapping IP addresses
- Anti-spoofing protection does function for packets originating from the shared internal interface. We recommend that you configure the internal physical router to perform anti-spoofing protection.

The Routing Concept section ("VSX Routing Concepts" on page 27) provides a detailed discussion of routing options in VSX environments.

Virtual Systems in the Bridge Mode

A Virtual System in the bridge mode implements native layer-2 bridging instead of IP routing. This allows network administrators to easily and transparently deploy a Virtual System in an existing network topology without reconfiguring the existing IP routing scheme. The figure below, shows a scenario where each Virtual System in the Bridge Mode protects a VLAN switched network.

Figure 11-35 Virtual Systems in Bridge Mode



Bridge Mode (on page 90) deployments are particularly suitable for large-scale clustered environments.

Cluster Deployments

This section presents several examples of cluster deployments that highlight important VSX features. The discussion is intended to introduce these features as they relate to deployment strategy. Refer to the conceptual discussion of cluster deployments ("Introduction to VSX Clusters" on page 82) section for more information.

Active/Standby Bridge Mode

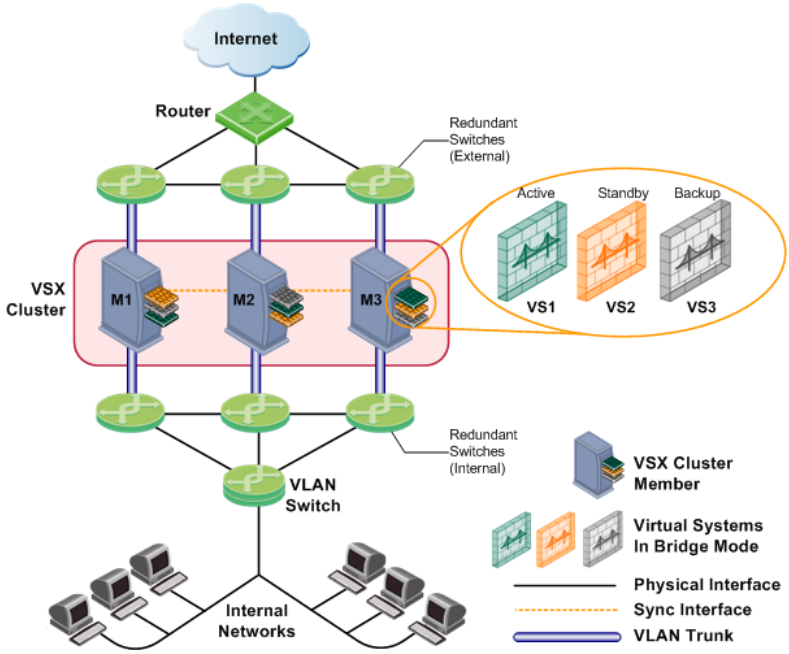
The Active/Standby Bridge Mode provides path redundancy and loop prevention, while offering seamless support for Virtual System Load Sharing and overcoming many Spanning Tree Protocol (STP) Bridge mode limitations.

The following sections describe two cluster deployment scenarios using the Active/Standby Bridge Mode.

VLAN Shared Interface Deployment - Active Standby Bridge Mode

In this scenario, each individual member connects to pair of redundant switches via a VLAN trunk. All Virtual Systems in a given member share the same VLAN trunk. The following figure illustrates an example of such a deployment with active, standby and backup members.

Figure 11-36 Active/Standby bridge mode - shared interfaces



When using the Active/Standby Bridge mode in a high availability deployment, VSX directs traffic to members according to predefined priorities and member status. In VSLS deployments, VSX distributes the traffic load amongst members according to a set of predefined preferences.

This deployment scenario is appropriate for very large enterprises.

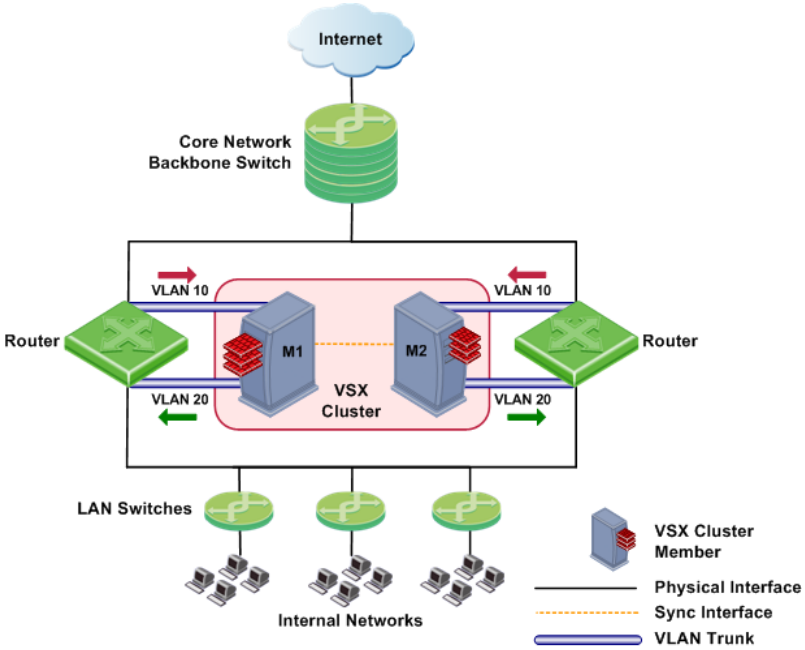
Virtual Systems in the Bridge Mode

A three layer hierarchical model is appropriate for large, high-traffic network environments. It contains a mixture of components as described below:

1. A **core network**, comprised of high-speed backbone switches directs traffic to and from the Internet and other external networks.
2. A **distribution layer**, comprised of routers, provides connectivity between the core and the access layer.
3. An **access layer**, comprised of redundant LAN switches, forwards traffic to and from internal networks.

VSX, using the Active/Standby Bridge mode, can be incorporated into the distribution layer, enforcing the security policy. This is illustrated in the following figure:

Figure 11-37 Active/Standby bridge mode - core network



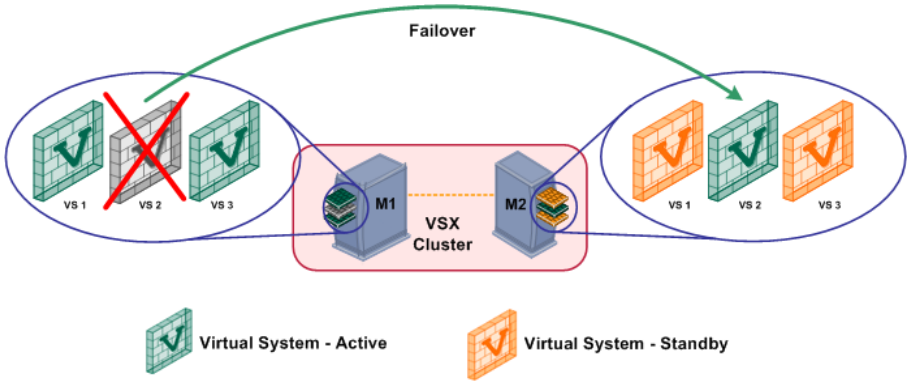
The routers direct external, "dirty" traffic (typically from the Internet) to the appropriate Virtual System via a segregated VLAN. Filtered, "clean" traffic exits the Virtual System via a separate segregated VLAN back to the routers and on to internal destinations.

This deployment scenario is appropriate for very large enterprises.

Per Virtual System High Availability

With per Virtual System high availability, each Virtual System monitors its own interfaces for failure.

Figure 11-38 Virtual System failover



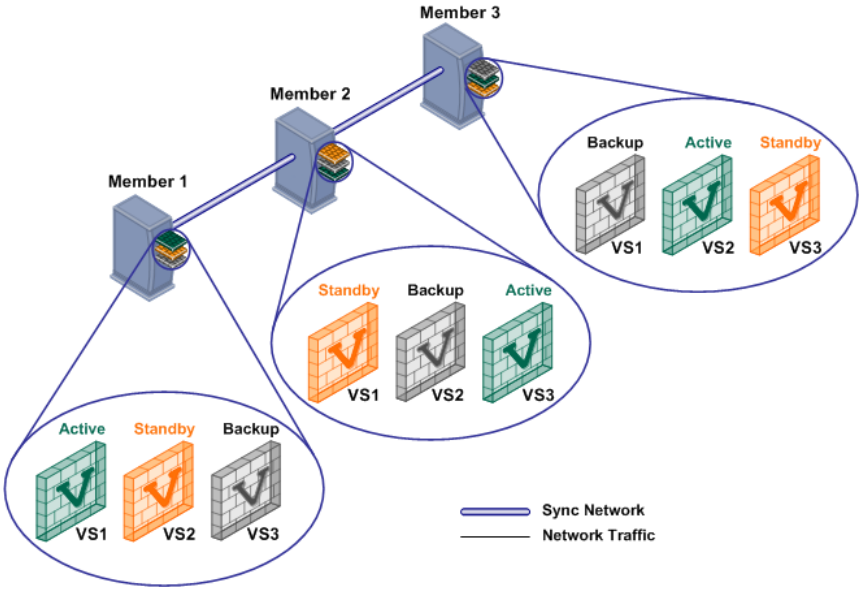
In this example, each cluster member contains three identical, synchronized Virtual Systems. The member designated as M1 currently process traffic for all Virtual Systems. If VS2 fails on M1, it fails over to its peer in M2. VS1 and VS3 continue to function normally on M1.

Virtual System Load Sharing (VSLS)

VSX clusters can efficiently balance network traffic load by distributing active virtual systems amongst cluster members. This capability is known as **Virtual System Load Sharing (VSLS)**.

The figure below illustrates a deployment scenario with three cluster members, each containing three Virtual Systems. In this configuration, an equalized load sharing deployment might have one active Virtual System on each cluster member.

Figure 11-39 Normalized VSLS deployment



A different member hosts the active peer for each Virtual System. This distribution spreads the load equally amongst the members. Once you create a Virtual System, VSX automatically assigns standby and backup states to the appropriate peers and distributes them among the other cluster members.

In the event that a cluster member fails, VSLS directs traffic destined to affected Virtual Systems to their fully synchronized standby peers, which then become active. At the same time, a backup Virtual Systems switches to standby, and synchronizes with the newly active Virtual System.

In the event that an individual active Virtual System fails, it immediately fails over to its standby peer and one of its backup peers becomes the standby, synchronizing with the newly active peer.

Organizational Deployment Strategies

This section presents deployment scenarios for different types of large organizations and illustrates how VSX provides security both internally and at the perimeter. The discussion covers the following types of organizations:

- Large Enterprises
- Managed Service Providers
- Data Centers

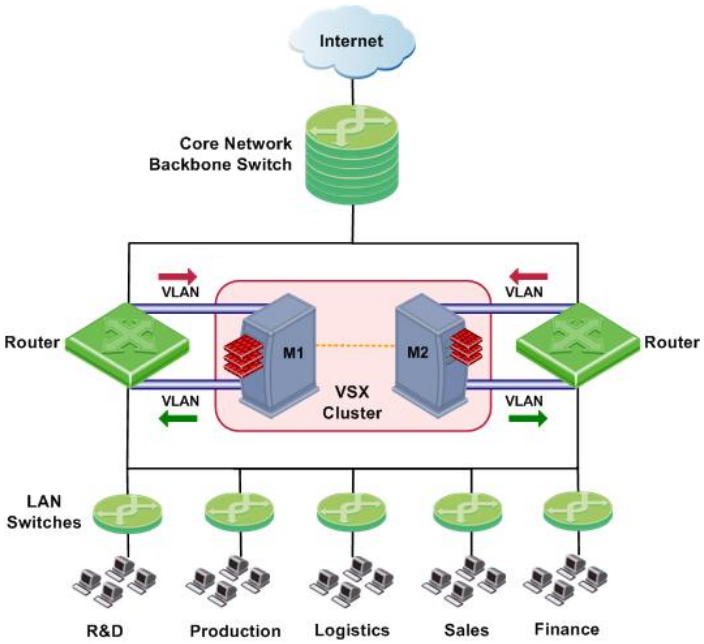
Enterprise Deployments

Large enterprise network environments typically contain a wide variety of diverse networks, distributed over multiple locations around the world. These networks often have differing security and access requirements for various departments and branches. The ability to centrally manage network security while maintaining throughput is a critical requirement.

Core Network Security

Many Enterprise environments are based on core networks. Situated adjacent to core network backbone switches, VSX protects the internal network by providing security at layer-2, layer-3 or both. VSX communicates with the core network using the existing infrastructure. With Virtual Systems in the Bridge Mode, VSX can protect departmental networks, while simultaneously preventing network segmentation. In this case, switches are located at the entrance to each department's network.

Figure 11-40 Core network deployment



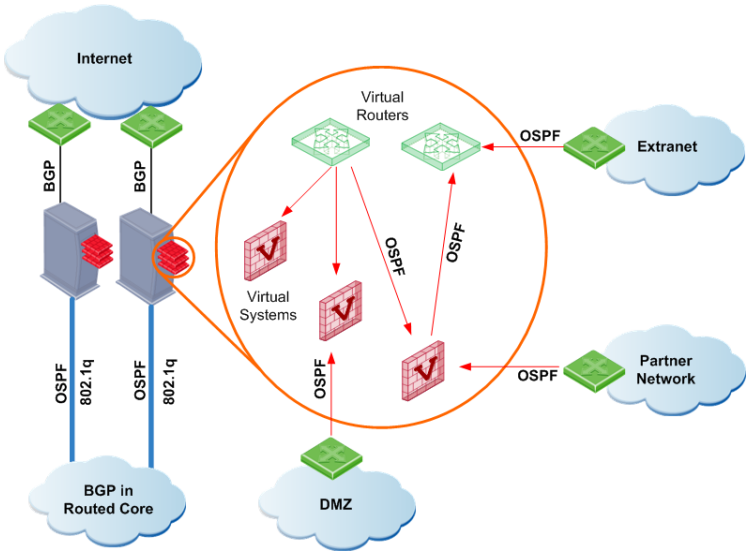
VSX ensures connectivity between the core network and the Internet or external networks, while providing perimeter security. Security can be configured on a per VLAN basis.

Dynamic Routing

The figure below presents a sample deployment of an enterprise network using dynamic routing protocols (OSPF/BGP). VSX secures each DMZ service, VPN peer, Domain and partner network while providing complete integration with dynamic routing protocols.

In this example, BGP neighbor updates in the routed core network are selectively redistributed to application networks. OSPF provides connectivity between Virtual Routers, Virtual Systems, the core network and application networks.

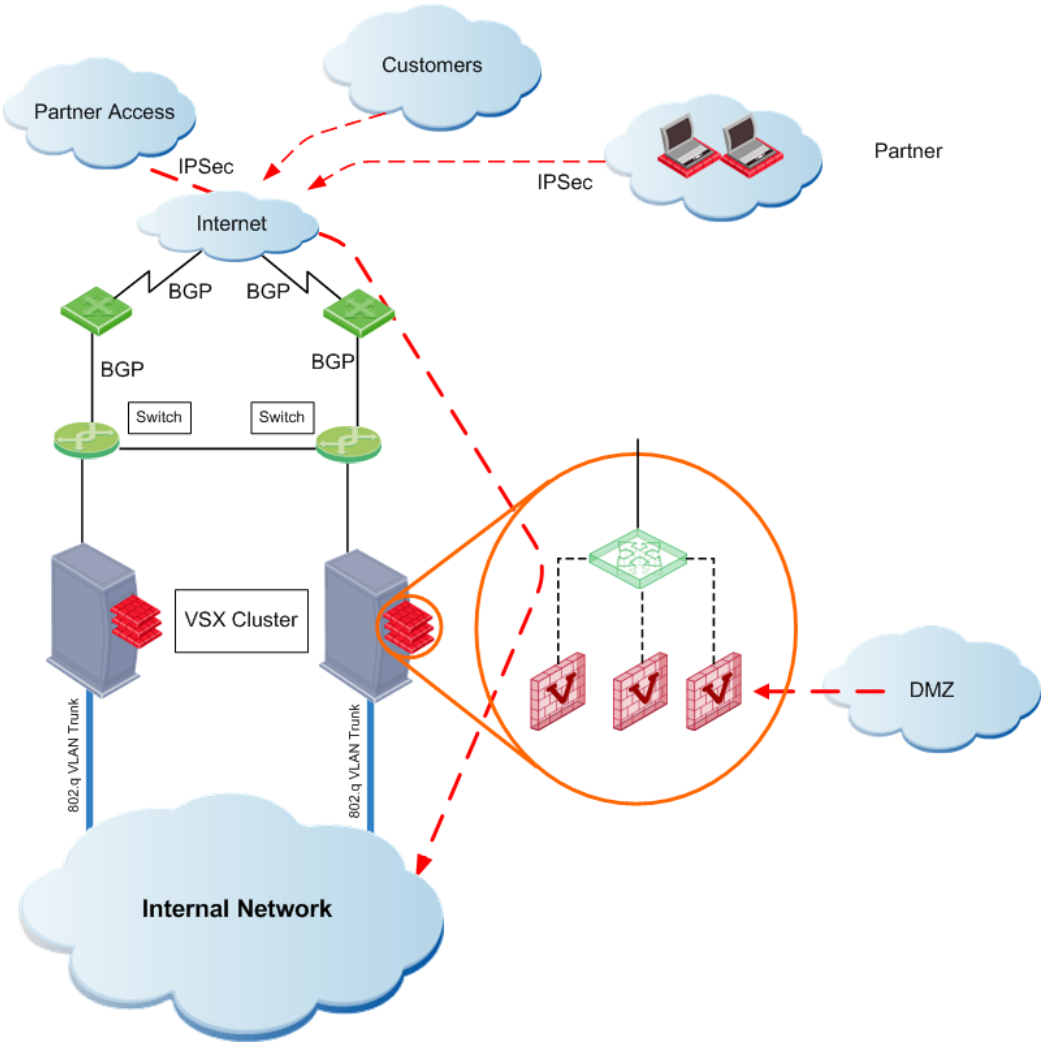
Figure 11-41 Dynamic routing



Perimeter Security

In the figure below, security is enforced on a per-VLAN basis. The OSPF and BGP Dynamic routing protocols provide connectivity to multiple security zones along the perimeter.

Figure 11-42 Perimeter security



Notes to this scenario:

- Partners access network resources remotely via Virtual Systems
- Each Virtual System has its own security policy based on its requirements
- Logs and audit information for each partner is collected separately, and saved to a private database
- Applications and services are segregated by private Virtual Systems
- Multiple Virtual Routers/Switches are used to control the access paths

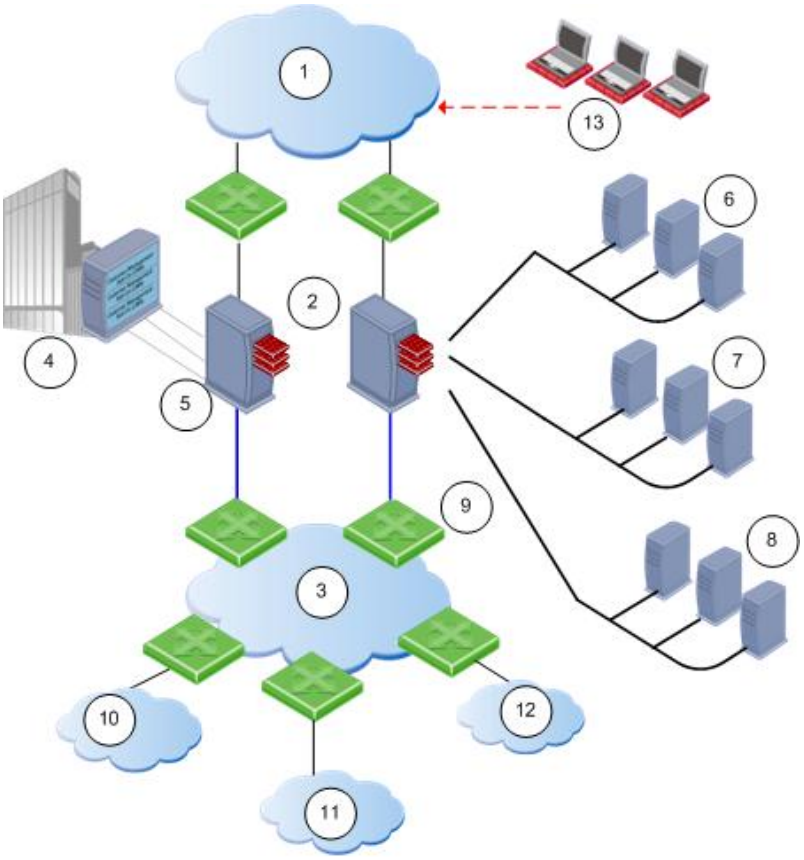
Managed Service Providers Using Multi-Domain Security Management

Managed service providers give connectivity and security services for Domain networks. Some of these Domains require remote access capabilities. In this service oriented environment, VSX and Multi-Domain Security Management provide central management and make connectivity and security easier, without affecting the existing IP topology.

In this scenario, a VSX cluster is in a Point of Presence (POP) deployment for a service provider. VSX consolidates hardware for the service provider and ensures privacy and secure connectivity solutions (VPN) for users. This scenario is appropriate for High Availability and Virtual System Load Sharing cluster modes.

VSX and Multi-Domain Security Management provide a centralized, granular provisioning system for a number of Domains. Applications and services are separated by discrete Virtual Systems. Access to these services and applications is based on need.

Figure 11-43 Multi-Domain Security Management Managed Service Provider



Scenario:

	Component Description
1	Internet. Routers are between the VSX cluster members and the Internet.
2	VSX cluster. One member handles the Local Exchange and another handles server traffic of different Domains.
3	Core IP VPN Network.
4	Multi-Domain Security Management at the Network Operation Center monitors POP and connects to VSX gateway. The Multi-Domain Log Server in the NOC collects data for each Domain and stores the logs in separate private databases.
5	Multi-Domain Security Management at the NOC and the VSX gateway make the Local Exchange.
6	Domain A web servers.
7	Domain B DMZ.
8	Domain C mail servers.
9	PE Router.
10, 11, 12	Domain A, B, and C. Each Domain manages its own security and cannot define Virtual Systems or other network components. Domains have secure VPN connectivity.

Data Centers

Data center providers supply external hosting services for Domain servers and databases. The service typically includes infrastructure, connectivity, and security for multiple Domains.

For example, you can have a scenario such as:

- Multiple Domain networks sharing a common physical infrastructure.
- Backbone that provides connectivity between each Domain and the data center.
- Domain A connects to its web hosting servers.
- Domain B connects to its mail servers.
- Domain C connects to its database servers.

To provide network security and management, the data center provider deploys a VSX gateway with one Virtual System for each Domain.

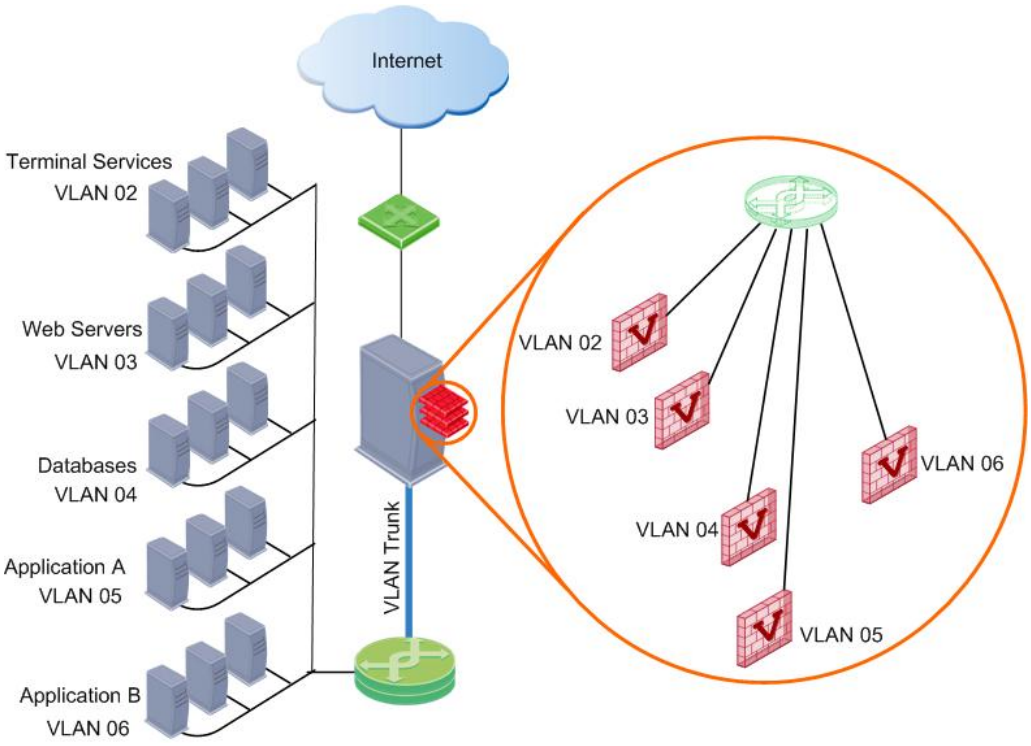
This scenario offers a cost effective scalability solution for network expansion by means of remote connectivity. In this example, a VPN connection between a Domain Virtual System and a UTM-1 Edge device protecting a remote network, integrates that network into the MPLS core. Similarly, a Virtual System can provide access for individual remote users who connect intermittently.

Data Centers in an Enterprise

This example scenario illustrates how VSX provides security management for enterprise data centers. By assigning layer-2 connections to Virtual Systems, VSX reduces the number of physically managed devices within a data center while providing the same high level of security.

In the figure below, a VSX gateway allows authorized users to access data center resources. The objective here is to protect shared resources with differing access permissions and security requirements, while implementing network granularity.

Figure 11-44 Enterprise data center



For example, one Virtual System protects databases against SQL vulnerabilities. Another Virtual System protects Web Servers using IPS. When new applications and services are added to the enterprise data center, new Virtual Systems are easily created to secure them according to their specific requirements.

Migrating from an Open Server to a VSX-1 Appliance

Check Point VSX-1 appliances use different interface names than Open Server platforms (SecurePlatform, Linux). When migrating an Open Server VSX gateway or cluster to a VSX-1 appliance, you must use the `vsx_util change_interfaces` command ("change_interfaces" on page 197) to change the appliance interface names.

The `vsx_util change_interfaces` command ("change_interfaces" on page 197) command contains a **Management Only** option that allows you to change the interface names on the management server (Security Management Server or Multi-Domain Security Management Domain Management Server) only. You then use the `vsx_util reconfigure` command ("reconfigure" on page 201) to push the updated configuration to VSX gateways or cluster members.

To migrate a VSX gateway or cluster to a VSX-1 appliance:

1. Close SmartDashboard for the Security Management Server and/or Multi-Domain Security Management Domain Management Servers.
2. On the management server, enter the **Expert Mode** and run the *vsx_util change_interfaces* command.

```
[Expert@mgt]# vsx_util change_interfaces

************************************************************************
***
* Note: the operation you are about to perform changes the information
in the management
* database. Back up the database before continuing.
************************************************************************
***

Enter SmartCenter Server/main Domain Management Server IP address (Hit
'ENTER' for 'localhost'): 172.23.5.151
Enter Administrator Name: aa
Enter Administrator Password:

************************************************************************
***
* It is highly recommended that all relevant Domain Management Servers
are unlocked during the entire operation
************************************************************************
***

Enter VSX Gateway/Cluster object name: MyCluster
```

3. When prompted, enter the Security Management Server, or Multi-Domain Security Management Main Domain Management Server IP address.
4. When prompted, enter the administrator name and password as prompted.
5. When prompted, enter the VSX cluster object name.
6. When prompted, select the **Management Only** option.

```
Change Interfaces can operate in two modes:
1. Apply changes to the management database and to the VSX
Gateway/Cluster members immediately
2. Apply changes to the management database only

* Choosing option 2 will require running 'vsx_util reconfigure'
* on a newly installed VSX Gateway/Cluster members after operation has
finished successfully

Please choose one of the above options (1|2) [1]:
```

```
Please select one of the following interfaces to be replaced:
  1)    lan0
  2)    lan1
Enter your choice:2

Please select one of the following interfaces to replace eth1:
  1)    A new interface name
Enter your choice:1

WARNING! Interface name must exist on VSX gateway/cluster members or
the operation will fail.

Enter new interface name: eth0
```

7. When prompted, select the interface to be replaced.
8. When prompted, enter **1** and then enter the new interface name .
9. When prompted to change another interface, enter "**y**" when prompted and repeat steps 7and 8 as required.
10. To complete the process, enter "**n**".
11. Open SmartDashboard and connect to the management server.
12. Open the VSX gateway or cluster object.
13. Select **Physical Interfaces** and select and remove all of the old interface names.
14. Click **OK** to complete the configuration.
15. On the VSX gateway or cluster members, run the vsx_util reconfigure ("reconfigure" on page 201) command.
16. Reboot the VSX gateway or cluster members.

Chapter 12

VSX Diagnostics and Troubleshooting

In This Chapter

Introduction	179
General Troubleshooting Steps	179
Troubleshooting Specific Problems	180

Introduction

This chapter presents basic diagnostic and troubleshooting procedures that should be followed in the event you encountering a problem while working with VSX. This diagnostic routine will assist you in determining the source of the problem. This chapter presents several known issues and their solutions.

Most problems are caused by configuration errors occurring during the process of defining VSX gateway, clusters and/or virtual devices. Another common source of problems involves networking and connectivity issues affecting VSX behavior. These problems are listed according to the order in which you will likely encounter them. Before reading and following a certain workaround, make sure you've read all the previous workarounds, and that those steps in the configuration were successful.

In some of the cases, one initial problem can cause problems in later stages of the configuration. For that reason, it is important to find the root of the problem when you are trying to understand what went wrong.

General Troubleshooting Steps

If you suspect that there is a problem with your VSX configuration, there are several diagnostic procedures that you can follow to determine the source. These procedures utilize various commands documented in the Command Line section ("Command Line Reference" on page 183).

1. Perform a basic configuration check for each gateway or cluster member by running the `fw vsx stat -v` command. The output will allow you to:
   a) Account for all Virtual Systems and verify that none are missing from the configuration.
   b) Verify that all Virtual devices are active
   c) Verify that the correct security policy is installed for each Virtual System
   d) Verify the SIC trust has been established with the management server
2. Run the `cplic` print command on each VSX gateway, cluster member and management server to verify that you have the appropriate licenses installed.
3. Run the `cphaprob stat` command on each cluster member to verify its status. If a member is listed with a status other than **Active**, **Standby, or Backup**, refer to the "*Troubleshooting*" chapter in the *R75.20 ClusterXL Administration Guide* (http://supportcontent.checkpoint.com/documentation_download?ID=12265) for additional troubleshooting assistance.
4. If you suspect that a Virtual System is experiencing connectivity problems, perform the following steps:
   a) Run: `vsx set` to set the context to the appropriate Virtual System.
   b) Run `fw [-v vsname | vsid] getifs` to display the interface list for the Virtual System.

c) Examine connectivity status using standard operating system commands and tools such as: `ping`, `traceroute`, `tcpdump`, `ip route`, `ftp`, etc. Some of these run according to context (i.e. routing, source and destination IP addresses). .

For SecurePlatform and Crossbeam platforms, execute the `ip route` and `ip link` commands**.**

If these tests indicate that all interfaces and routers have connectivity, and appear to be functioning correctly, you should monitor the passage of packets through the system.

5. Execute the `fw monitor -v [vsname or vsid]` commands to capture details of packets at multiple points. This may return multiple reports on the same packet as it passes various capture points. This command does not report on Virtual Routers, except for packets destined to an external Virtual Router.

Note - The Performance Pack may have an adverse effect on the capabilities of the `fw monitor` command.

6. Execute the `tcpdump` command to display transmitted or received packets for specific interfaces, including Warp interfaces. This often provides valuable clues for resolving connectivity issues.

Troubleshooting Specific Problems

Cannot Establish SIC Trust for Gateway or Cluster

When creating a VSX gateway or cluster, you cannot establish SIC trust. SmartDashboard gives an error message:

Certificate cannot be pushed. Connection error with wait agent.

Possible Causes	How to Resolve
Check that you have network connectivity between the gateway and the Security Gateway or Domain Management Server by pinging from the VSX system (A ping from the Domain Management Server/Security Management to the VSX system will not work because of the default security policy installed on the VSX gateway/cluster.) **Make sure the context is vrf 0 first**.	On all relevant machines, re-check the cables, routes, IP addresses and any intermediate networking devices (routers, switches, hubs, and so on) between the management and the gateway(s).
Check that all the Check Point processes on the VSX gateway(s) are up and running by running `cpwd_admin list` and making sure each line has a non-zero value in the **PID** field.	If the gateway(s) has just rebooted, the Check Point processes might still be coming up. If this is not the case, and you are using Crossbeam X40, make sure you have executed the `application … start` command. (For more information refer to the Crossbeam documentation.)
Check that the CPD process is listening to the trust establishment port, by running `netstat -an | grep 18211` on the VSX gateway(s), and checking that output looks like this: `tcp 0 0 0.0.0.0:18211 0.0 .0.0:* LISTEN`	Make sure that you executed the `cpconfig` command on the gateway(s), and that it finished successfully.

SIC Trust Problems with new Virtual Devices

When creating a new Virtual System, Virtual Router or Virtual Switch, you cannot establish SIC trust.

Possible Causes	How to Resolve
Time or time zone mismatch between the management and the gateway. For proper SIC operation, the time, date and time zone must be synchronized between the management server and gateways/ cluster members. Execute the `/bin/date -u` command on all machines, to obtain the correct UTC/GMT time. The machines can be in different time zones, as long as their UTC/GMT times match.	Change the time, date and time zone on the management and/or the gateway(s) so that their UTC/GMT times match. Refer to you operating system documentation for the exact commands needed to accomplish this.

Re-establishing SIC Trust with Virtual Devices

In the event that you encounter connectivity problems due to the loss of SIC trust for a specific virtual device, you can use the following procedure to manually re-establish trust.

To manually re-establish SIC Trust with virtual devices:

1. Execute the following command from the VSX Gateway command line (In the expert mode): `vsx sic reset <vsid>`.

   `vsid`: Identification number of the virtual device
2. Execute the following command(s) on the management server:

   a) `# MDSsnv <target_domain_name>`

   (Multi-Domain Security Management only)

   b) `# cpca_client revoke_cert -n <vs_sic_name>`

   vs_sic_name: Virtual device SIC name. To determine the SIC name, run `guidbedit.exe` and search for the sic_name attribute on the virtual device network object.
3. In SmartDashboard, open the virtual device object and click **OK**. This action creates a new SIC certificate for the virtual device and saves it on the VSX gateway.

Sync Networks Do Not match

If you click **Get Configuration** in the **VSX Cluster Creation** wizard and an error message shows that the sync networks do not match, see if the IP addresses on the sync interface are valid. The Addresses must be unique for each member, must belong to the same network and must have the same Net mask. The interface that is used for Sync was defined during the initial configuration phase on the VSX gateway.

To resolve, change the IP address/Net mask of the Sync interface, restart the VSX gateway(s), and click **Get Configuration** again.

Install Policy Error Using VSX Creation Wizard

After completing the VSX creation wizard, a failure occurs and the following message appears in the **Operation Report** window: `Error: Default policy installation failed on VSX. Install policy manually using` SmartDashboard.

Possible Causes	How to Resolve
Missing or invalid license on the management server. Execute `cplic check` on the management server to verify that you have the required licenses.	Obtain and install the appropriate licenses.

Possible Causes	How to Resolve
Missing or invalid VSX gateway/cluster licenses. Run `fw vsx stat` on all gateways, and make sure that the output says **Number of Virtual Systems allowed by license:** is greater than **0**.	Obtain a VSX and install a valid license for each VSX gateway or cluster member.
Time or time zone mismatch between the management and the gateway. For proper SIC operation, the time, date and time zone must be synchronized between the management server and gateways/ cluster members. Execute the `/bin/date -u` command on all machines, to obtain the correct UTC/GMT time. The machines can be in different time zones, as long as their UTC/GMT times match.	Change the time, date and time zone on the management and/or the gateway(s) so that their UTC/GMT times match. Refer to you operating system documentation for the exact commands needed to accomplish this.

Internal Host Cannot Ping Virtual System

After defining a Virtual System with an internal VLAN interface, an internal host on that VLAN cannot ping the Virtual System internal or external IP address.

Possible Causes	How to Resolve
A policy allowing the communication was not installed on the Virtual System. Note that after creating a Virtual System, it has a default policy that blocks all traffic.	Install a policy on the Virtual System that enables the traffic. Check with the SmartView Tracker that the Virtual System is allowing the traffic.
There is the VLAN configuration problem on a switch, or physical cable problem.	Check the switch configuration. Make sure that VLAN tag configured on the switch is the same as used for the Virtual System VLAN interface. Check the cables, and make sure that you have plugged the cable from the switch to the correct port on the VSX gateway or cluster members.
Incorrect routing on adjacent routers or hosts.	Check the routing tables on intermediate routers and hosts. You can use `tcpdump` on the relevant VLAN interface on the VSX gateway or cluster member to verify that the traffic is arriving to and leaving the VSX machine.
Incorrect IP address or net mask defined on the Virtual System VLAN interface.	Check the IP address and the net mask assigned to the Virtual System internal VLAN interface.

Chapter 13

Command Line Reference

In This Chapter

Firewall Commands	183
VSX Command	187
Link Aggregation CLI Commands	191
VSX Resource Control Commands	192
The vsx_util Command	195
The cphaprob Command	206

Firewall Commands

This section presents the usage of standard firewall (fw) commands as applicable to VSX gateways and Virtual Systems.

fw getifs

Description Displays `a driver interface list for a specific` Virtual System. By default, the VSX gateway interface is displayed.

Syntax

```
fw [-vs vsid | vsname] getifs
```

Parameters

Parameter	Description
`[-vs vsid | vsname]`	Specify a Virtual System by name or ID. If you omit the `-vs` argument, only those interfaces associated with the current context appear.

Return Value `0` (zero) indicates that the command executed successfully. Any other response indicates an error.

Output

```
fw getifs
localhost vnd0 0.0.0.0 0.0.0.0
localhost eth0 4.4.6.101 255.255.0.0
localhost eth1 0.0.0.0 0.0.0.0
localhost sdp0 0.0.0.0 0.0.0.0
localhost int 0.0.0.0 0.0.0.0
localhost mgmt 10.18.83.171 255.255.255.0
localhost sync 7.7.7.171 255.255.255.0
localhost wrpj50001 0.0.0.0 0.0.0.0
localhost wrpj50003 0.0.0.0 0.0.0.0
```

fw monitor

Description Captures network packets at multiple points within the VSX environment. You can only run one instance of this command at a time on VSX gateway.

This section only presents the syntax relevant for VSX gateways or clusters.

Syntax

```
fw monitor [-v vsid]
```

Parameters

Parameter	Description
`[-v vsid]`	Specify a gateway or Virtual System by its ID. the specific Virtual System on which packets should be captured. The default gives the VSX gateway.

Return Value `0` (zero) indicates that the command executed successfully. Any other response indicates an error.

Example `fw monitor -v 2 -e 'accept ip_p=6` shows all TCP packets passing through Virtual System 2.

Output

```
[member1:0]# fw monitor
 monitor: getting filter (from command line)
 monitor: compiling
monitorfilter:
Compiled OK.
 monitor: loading
 monitor: monitoring (control-C to stop)
eth4:o[124]: 192.168.200.171 -> 192.168.200.1 (TCP)
len=124 id=56430
TCP: 22 -> 1794 ...PA. seq=28d95f71 ack=57e454b1
eth4:O[124]: 192.168.200.171 -> 192.168.200.1 (TCP)
len=124 id=56430
TCP: 22 -> 1794 ...PA. seq=28d95f71 ack=57e454b1
eth4:i[40]: 192.168.200.1 -> 192.168.200.171 (TCP)
len=40 id=64876
TCP: 1794 -> 22 ....A. seq=57e454b1 ack=28d95fc5
eth4:I[40]: 192.168.200.1 -> 192.168.200.171 (TCP)
len=40 id=64876
TCP: 1794 -> 22 ....A. seq=57e454b1 ack=28d95fc5
monitor: caught sig 2
monitor: unloading
```

fw tab

Description Displays state tables for a specific Virtual System. State tables are used to store state information that Virtual Systems use to correctly inspect packets.

Syntax

```
fw [-vs vsid | vsname] tab [-t table name] [...]
```

Description Displays state tables for a specific Virtual System. State tables are used to store state information that Virtual Systems use to correctly inspect packets.

Parameters

Parameter	Description
`[-vs vsid | vsname]`	Specify a Virtual System by name or by ID. If no value is specified, the VSX gateway is assumed.
`-t table name`	Shows the state table for the specified Virtual System.
`[...]`	Arguments as defined for non-VSX machines.

Example

```
fw -vs 1 tab -t connections
```

Output

```
localhost:
-------- connections --------
dynamic, id 8158, attributes: keep, sync, expires
25, refresh, limit 15000, hashsize 32768, kbuf 16 17
18 19 20 21 22 23 24 25 26 27 28 29 30, free
function 90adc508 0

00000000, 0a125364, 00008a69, 0a12ae0a, 00004710,
00000006; 0001c001, 00804000, 08000000, 00000e10,
00000000, 3f7c2df6, 00000001, 66010202, 000007b6,
ffffffff, ffffffff, ffffffff, ffffffff, 00000000,
00000000, 00000000, 00000000, 00000000, 3bcd7000,
00000000, 00000000, 00000000; 3596/3600>
00000001, 0a12ae0a, 00004710, 0a125364, 00008a69,
00000006> 00000000, 0a125364, 00008a69, 0a12ae0a,
00004710, 00000006> 00000005)
00000000, 0a125364, 00008a6c, 0a12ab0a, 00004710,
00000006; 0001c001, 00804000, 08000000, 00000e10,
00000000, 3f7c2e4b, 00000001, 65060404, 000007b6,
0000000b, 0000000b, ffffffff, ffffffff, 00000000,
00000000, 00000000, 00000000, 00000000, 1fce4000,
00000000, 00000000, 00000000, 00000000, 00000000;
3581/3600>
00000001, 0a12ac0a, 00004710, 0a125364, 00008a6b,
00000006> 00000000, 0a125364, 00008a6b, 0a12ac0a,
00004710, 00000006> (00000005)
```

fw fetch

Description Fetches the Inspection Code from the specified host and installs it to the kernel.

Usage `fw fetch [-n] [-f <filename>] [-c] [-i] master1 [master2] ...`

Syntax

Argument	Description
`-n`	Fetch the Security Policy from the Security Management Server to the local `state` directory, and install the Policy only if the fetched Policy is different from the Policy already installed.
`-f <filename>`	Fetch the Security Policy from the Security Management Server listed in <filename>. If filename is not specified, the list in `conf/masters` is used.

Argument	Description
`-c`	Cluster mode, get policy from one of the cluster members, from the Check Point High Availability (CPHA) kernel list.
`-i`	Ignore SIC information (for example, SIC name) in the database and use the information in `conf/masters`. This option is used when a Security Policy is fetched for the first time by a DAIP gateway from a Security Management Server with a changed SIC name.
`master1`	Runs the command on the designated master. The name of the Security Management Server from which to fetch the Policy. You may specify a list of one or more Security Management Servers, such as `master1 master2` which will be searched in the order listed. If `targets` is not specified, or inaccessible, the policy is fetched from `localhost`.
-vs <VSID>	Fetches a Security Policy to the specified Virtual System.

VSX Command

This section describes the vsx commands.

Note - `fw6 vsx` commands are not supported. Because all IPv6 commands require a corresponding IPv4 connection, `fw6 vsx` commands are not necessary.

vsx fetch

Description

```
Fetches the most current configuration files from
the Main Domain Management Server, and applies it to
the VSX gateway.
```

Syntax

```
vsx fetch [-v] [-q] [-s] local
vsx fetch [-v | -q| -s] [-f conf_file]
vsx fetch [-v | -q] -C "command"
vsx fetch [-v | -q| -c| -n| -s] [management]
```

Parameters

Parameter	Description
`-c`	Cluster mode
`-n`	Do not run `local.vsall` if VSX configuration, as fetched from management server, is up-to-date.
`-s`	Concurrent fetches for multi-processor environment.
`-q`	Quiet mode - Only summary lines appear.
`-v`	Verbose mode - Detailed information appears.
`-f conf_file`	Fetches NCS commands configuration file instead of the default `local.vsall`.
`local`	Reads `local.vsall` configuration file from `$FWDIR/state/local/vsx` and executes the NCS
`management`	Fetches `local.vsall` from management, replaces and runs it.
`-C command`	Execute NCS command

Return Value

`0` (zero) indicates that the command executed successfully. Any other response indicates an error.

Description Fetches the most current configuration files from the Main Domain Management Server, and applies it to the VSX gateway.

Output

```
fw vsx fetch
Fetching VSX Configuration From: 10.18.99.101

 Local VSX Configuration is Up-To-Date.
 Cleaning un-used Virtual Systems entries
(local.vskeep).

 Purge operation succeeded.
 Fetching Virtual Systems configuration file
(local.vsall).

SecureXL device has been enabled for vsid 1
SecureXL device has been enabled for vsid 2
SecureXL device has been enabled for vsid 3
Virtual Systems configuration file installed
successfully
```

vsx fetchvs

Description Retrieves a specific Virtual System configuration file based on information stored locally on the gateway.

Syntax

```
vsx fetchvs [-v | -q] [VS name | vsname]
```

Parameters

Parameter	Description
-q	Quiet output. Only summary information appears.
-v	Verbose output. Detailed information appears.
VS Name|vsid	Enter the Virtual System name or ID.

Return Value 0 (zero) indicates that the command executed successfully. Any other response indicates an error.

Example

```
fw vsx fetchvs California
```

Output

```
fw vsx fetchvs 2
SecureXL device has been enabled for vsid 2
```

vsx get

Description Returns the current shell context.

Syntax

```
vsx get
```

Output

```
vsx get
Current context is Virtual Device VS2 (ID 3).e
```

vsx set

Description Sets current context to the specified Virtual System by name or ID.

Syntax

```
vsx set [VSname | vsid]
```

Description Sets current context to the specified Virtual System by name or ID.

Parameters

Parameter	Description
`VSname or vsid`	Virtual System name or ID. If no value is entered, the context is set to the VSX gateway.

Return Value `0` (zero) indicates that the command executed successfully. Any other response indicates an error.

Example

```
vsx set 2
```

Output

```
vsx set 2
Context is set to Virtual Device VS1 (ID 2)
```

Comments This command can be used to check the connectivity between a specific Virtual System and physical or Virtual Routers. The command line prompt also displays the current context.

vsx stat

Description Displays VSX status information.

Syntax

```
vsx stat [-v] [-l] [vsid]
```

Parameters

Parameter	Description
`-v`	Displays detailed (verbose) information.
`-l`	Displays a detailed list of all virtual devices.
`vsid`	Displays statistics for the specified Virtual System

Output

Description Displays VSX status information.

```
VSX Gateway Status
==================
Name:                 MyGateway
Security Policy:       MyGateway_VSX
Installed at:          10Dec2007 10:31:25
SIC Status:            Trust

Number of Virtual Systems allowed by license:100
Virtual Systems [active/configured]: 2/2
Virtual Routers and Switches [active/configured]:1/1
Total connections [current/limit} 4/46000

Virtual Devices Status
======================
 ID   | Type & Name | Security Policy | Installed at   | SIC
------+-------------+-----------------+----------------+------
   1  | W VSW_1     |<Not Applicable> |                | Trust
   2  | S VS1       |Standard         | 10Apr2005 10:31| Trust
   3  | S VSs2      |Standard         | 10Apr2005 10:31| Trust

Type: S - Virtual System, B - Virtual System in Bridge mode,
      R - Virtual Router, W - Virtual Switch,
      ? - Information unavailable.
```

Output (for a specific Virtual System)

```
[Expert@sun:51]# vsx stat -l 51

VSID:            51
VRID:            51
Type:            Virtual System
Name:            sun_mem_vs49
Security Policy: TDCH-FW08.OPA
Installed at:    29Jul2009 12:37:58
SIC Status:      Trust
Connections number: 0
Connections peak:   0
Connections limit:  15000
```

vsx start_dr

Description Enables dynamic routing on Virtual Systems and previously configured Virtual Routers

Syntax

```
vsx start_dr
```

Output

```
[Expert@test:0]# vsx start_dr
Successfully activated dynamic routing on all Virtual
Systems/Routers
[Expert@test:0]#
```

vsx sic reset

Description Resets SIC for the specified Virtual System

Syntax

```
vsx sic reset {vsname|vsid}
```

Parameters

Parameter	Description
vsname|vsid	Specify the Virtual System name or ID.

Output

```
[Expert@gateway:0]#
[Expert@gateway:0]# fw vsx sic reset 1
resetting SIC for VSID 1
[Expert@gateway:0]#
```

Note - On the management server, use the `cpca_client revoke_cert` command to cancel the old certificate. In SmartDashboard, open the Virtual System object for editing. Click **OK**. This action creates a new certificate, and transfers the certificate to the gateway.

Link Aggregation CLI Commands

cphaconf show_bond

Displays the status of an interface bond, or with the -a argument, a summary table of all bonds. When a bond is specified information for each slave interface is also displayed.

Syntax

```
cphaconf show_bond < <bond-name>|-a>
```

Example

```
[Expert@GW-1]# cphaconf show_bond bond0

Bond name:       bond0
Bond mode:       Load Sharing
Bond status:     Up
Balancing mode:  802.3ad Layer3+4 Load Balancing
Configured slave interfaces: 4
In use slave interfaces:     4
Required slave interfaces:   2

Slave Name       | Status          | Link
-----------------------------------------
eth2             | Active          | Yes
eth3             | Active          | Yes
eth4             | Active          | Yes
eth5             | Active          | Yes
```

Report Results

- `Required slave interfaces` as explained in ("Setting Critical Required Interfaces" on page 139).
- The `Status` column can contain the following values:
  - `Down` (Load Sharing mode only) - the physical link is down.
  - `Active` - currently handling traffic.
  - `Standby` (High Availability mode only) - the interface is ready, and can support internal bond failover.
  - `Not Available` (High Availability mode only) - either the physical link is broken, or that the Cluster member is in status *down*. The bond cannot failover in this state.
- The `Link` column reports whether the physical link exists.

chpaconf failover_bond

Initiates bond interface failover in the High Availability mode.

Syntax

```
cphaconf failover_bond <bond-name>
```

cphaprob -a if

Displays the status of all interface bonds and VLANs. For an HA bond, specifies whether it can failover.

Syntax

```
cphaprob -a if
```

Example

```
[Expert@GW-1]# cphaprob -a if

Required interfaces: 5
Required secured interfaces: 1

bond0  UP      non sync(non secured), broadcast, bond, can
failover
bond2  UP      sync(secured), multicast, bond Load Sharing
bond1  UP      non sync(non secured), multicast, bond Load
Sharing

Virtual cluster interfaces: 4

bond0      192.168.34.60
bond1.60   10.34.60.1
bond1.61   10.34.61.1
bond1.62   10.34.62.1
```

VSX Resource Control Commands

The following commands work with the VSX Resource Control feature.

vsx resctrl enforce

Description Configures the Resource Control Enforcer and shows its current status. This command overrides the settings in the Resource Control configuration file, but does not survive reboot.

Syntax

```
vsx resctrl enforce {enable | disable | show}
```

Parameters

Parameter	Description
`enable`	Enables Resource Enforcer
`disable`	Disables Resource Enforcer
`show`	Displays whether Resource Enforcer is enabled or disabled

vsx resctrl monitor

Description Configures the Resource Monitor and shows its current status. This command overrides the settings in the Resource Control configuration file, but does not survive reboot.

Syntax

```
vsx resctrl monitor {enable | disable | show}
```

Parameters

Parameter	Description
`enable`	Enables Resource Monitor
`disable`	Disables Resource Monitor
`show`	Displays whether Resource Monitor is enabled or disabled

vsx resctrl traffic_stat

Description Displays the number of received and dropped packets for a Virtual System. The counter can be reset using the `vsx resctrl reset` command.

Syntax

```
vsx resctrl -w {vsid} traffic_stat
```

Parameters

Parameter	Description
`-w {vsid}`	Displays the statistics for the Virtual System specified in {vsid}. The default vsid is 0.

vsx resctrl reset

Description Resets the Resource Control monitoring statistics.

Syntax

```
vsx resctrl reset
```

vsx resctrl start

Description Initializes Resource Control. Use this command after changing the weights of the Virtual Systems in the configuration file.

Syntax

```
vsx resctrl [-v] start
```

Parameters

Parameter	Description
-v	Verbose mode displays the configuration of the Resource Control Monitor and the Resource Control Enforcer during initialization.

vsx resctrl stat

Description Displays CPU consumption per-Virtual System, and serves as an accurate indicator of the total CPU consumption by all Virtual Systems. CPU consumption is expressed as a percentage of the total CPU resources.

Syntax

```
vsx resctrl [-u | -q] stat
```

Parameters

Parameter	Description
-u	Displays information per PU (SMP only).
-q	Do not display header row

Output Without Arguments

```
Virtual Systems CPU Usage Statistics
====================================

Number of CPUs/Hyper-threading: 4
Monitoring active time: 14s

ID    Name       |Weight|  1sec  10sec   1min    1hr   24hr*
================+======+=================================
  0   VSX2       |  N/A |  0.11   0.06   0.08   0.07   0.01
  1   VSX2_vs1   |  10  | 15.80  21.57  21.75  22.28   1.94
  2   VSX2_vsw   |  N/A |  0.00   0.00   0.00   0.00   0.00
  3   VSX2_vs2   |  10  | 16.91  22.57  22.77  23.09   2.01
================+======+=================================
Total VS CPU Usage     | 32.82  44.20  44.60  45.44   3.96
======================+=================================
System CPU Usage       |    99     99     99     99     89
```

Output with -u argument

```
Virtual Systems CPU Usage Statistics
====================================

Number of CPUs/Hyper-threading: 2
Monitoring active time: 14s

ID  Name    | Weight | CPU   | 1sec   10sec  1min   1hr    24hr
==========+========+=======+================================
0   rescon |  N/A   | CPU 0 | 0.35   0.27   0.29   0.06   0.04
                    | CPU 1 | 1.23   0.33   0.33   0.16   0.22
                    | AVG   | 0.35   0.30   0.31   0.11   0.13
----------+--------+-------+--------------------------------
1   VS1    |   25   | CPU 0 | 0.63   0.58   0.63   0.05   0.00
                    | CPU 1 | 0.32   0.21   0.19   0.07   0.04
                    | AVG   | 0.48   0.40   0.41   0.06   0.02
----------+--------+-------+--------------------------------
2   VS1    |   40   | CPU 0 | 0.94   0.95   1.00   0.12   0.06
                    | CPU 1 | 0.00   0.00   0.00   0.00   0.00
                    | AVG   | 0.94   0.95   1.00   0.12   0.06
----------+--------+-------+--------------------------------
3   VS2    |   15   | CPU 0 | 0.35   0.34   0.18   0.02   0.06
                    | CPU 1 | 0.00   0.00   0.00   0.00   0.00
                    | AVG   | 0.35   0.35   0.18   0.02   0.06
==================+=======+================================
Total Virtual Dev   | CPU 0 | 7.11   2.25   3.80   0.89   0.45
CPU Usage           | CPU 1 | 3.00   5.10   2.32   1.02   0.30
                    | AVG   | 5.06   3.68   3.06   0.96   0.38
```

Comments

- For systems with more than one CPU, the time displayed is an average for all CPUs. To see the usage for each Virtual System per CPU, execute `vsx resctrl -u stat`.
- The VSX gateway itself, Virtual Switches and Virtual Routers are not assigned weights because they always receive the highest priority.
- Total Virtual System CPU Usage represents the total CPU utilization for all virtual devices, including Virtual Routers, Virtual Switches and the VSX gateway.
- System CPU Usage reports the total CPU utilization for the entire machine.

The vsx_util Command

Description Performs various VSX maintenance tasks. You run this command from the expert mode on the management server (Security Management Server or a Main Domain Management Server in a Multi-Domain Security Management environment).

Syntax

```
vsx_util <sub-command> [parameters]
```

Parameters

Parameter	Description
`-s <management IP>`	Perform action using the specified management IP
`-u <user name>`	Perform the action using the specified administrator
`-c <cluster name>`	Perform the action on the specified cluster
`-m <member name>`	Perform the action on the specified member
`-h`	Display help text

Description Performs various VSX maintenance tasks. You run this command from the expert mode on the management server (Security Management Server or a Main Domain Management Server in a Multi-Domain Security Management environment).

Comments **Note** - You must close SmartDashboard before executing the `vsx_util` command if any Virtual Systems are defined on the Security Management Server or Multi-Domain Security Management Domain Management Server. Failure to do so may result in a database locked error.

The `vsx_util` command typically requires you to enter the following information before executing the command:

- Management server name or IP address
- User name and password
- The command may ask for the name of one or more VSX objects upon which the command operates
- Most vsx_util sub-commands are interactive and require additional user input. Brief descriptions of additional input requirements appear in the Input section for the various sub-commands. The instructions on the screen typically provide helpful information regarding required information.

vsx_util Subcommands

add_member 196
add_member_reconf 197
change_interfaces 197
change_mgmt_ip 198
change_mgmt_private_net 198
fw fetch 199
change_interfaces 199
change_mgmt_subnet 201
convert_cluster 201
reconfigure 201
remove_member 202
show_interfaces 202
upgrade 203
view_vs_conf 203
vsls 205

add_member

Description Adds a new member to an existing VSX cluster.

Syntax
```
vsx_util add_member
```

Input
- VSX cluster object name
- New member name
- IP for [interface]: IP address assigned to specified interface (IP address is required for management and sync network interfaces)

Comments Run the command and follow the instructions on the screen. When the command finishes executing, you must also Run the vsx_util add_member_reconf command.

See Adding a New Member (on page 107) before using this command.

add_member_reconf

Description Restores VSX configuration after adding a cluster member

Syntax
```
vsx_util add_member_reconf
```

Input
- VSX member object name: VSX cluster member name
- Activation Key: SIC activation key assigned to the Security Management Server or main Domain Management Server
- Retype Activation Key: Retype to confirm the SIC activation key

Comments Execute the command and follow the instructions on the screen. Reboot the member after the command script finishes.

Review the procedure for defining a new member (See "Adding a New Member" on page 107) before using this command.

change_interfaces

Description Automatically replaces designated existing interfaces with new interfaces on all virtual devices to which the existing interfaces connect.

This command is useful when converting a deployment to use Link Aggregation, especially where VLANs connect to many virtual devices.

Syntax
```
vsx_util change_interfaces
```

Comments
- This command is interactive. Follow the instructions on the screen.
- This command supports the resume feature.
- You can use this command to migrate a VSX deployment ("Migrating from an Open Server to a VSX-1 Appliance" on page 177) from an Open Server to a Check Point appliance by using the **Management Only** mode.
- Refer to the notes ("Notes" on page 198) for additional information.

Important - You must close SmartDashboard for all Multi-Domain Security Management Domain Management Servers using the affected interfaces prior to running this command.

Using vsx_util change_interfaces

To change interfaces:

1. Close SmartDashboard for the Security Management Server and/or Multi-Domain Security Management Domain Management Servers.
2. On the management server, enter the Expert Mode and run the *vsx_util change_interfaces* command.
3. Enter the Security Management Server or Multi-Domain Security Management main Domain Management Server IP address.
4. Enter the administrator name and password as requested.
5. Enter the VSX cluster object name.
6. When prompted, select one of the following options:
   - **Apply changes to management and Security Gateway/cluster members**: Changes the on the management server, the VSX Security Gateway and cluster members.
   - **Apply changes to management Only**: Changes interface on the management server only. You must use the `vsx_util reconfigure command ("reconfigure" on page 201)` to push the updated configuration to VSX gateways or cluster members.

7. When prompted, select the interface to be replaced.
8. When prompted, select the replacement interface.
   a) You can optionally add a new interface by selecting "**Enter new interface name**". This interface must physically exist on the VSX Gateway or cluster members or the operation will fail.
   b) At the prompt, enter the new interface name. If the new interface is a bond, the interface name must match the bond name exactly (bond names are case sensitive).
9. To replace additional interfaces, enter "**y**" when prompted and repeat steps 6 through 8.
10. To complete the process, enter "**n**".
11. If you selected the **Apply changes to management only** option, run the `vsx_util reconfigure` command ("reconfigure" on page 201) to push the updated configuration to the VSX gateways or cluster members.
12. Reboot the VSX gateway and/or cluster members as appropriate.

Notes

- The **Apply changes to management and gateway/cluster members** option verifies connectivity between the management server and the VSX gateway or cluster members. In the event of a connectivity failure one of the following actions occur:
  a) If **all** of the newly changed interfaces fail to establish connectivity, the process terminates unsuccessfully.
  b) If one or more interfaces successfully establish connectivity, while one or more other interfaces fail, you may optionally continue the process. In this case those interfaces for which connectivity was established successfully will be changed. For those interfaces that failed, you must then resolve the issue and then run the vsx_util reconfigure ("reconfigure" on page 201) command to complete the process.
- If you select the **Apply changes to management Only** option, you can select another interface from list (if any are available) or select the option to add a new interface.

change_mgmt_ip

Description Changes gateway or cluster member management IP address

Syntax
```
vsx_util change_mgmt_ip
```

Input
- VSX gateway/member object name
- New management IP address

Comments
- We recommend that you back up the management database before using this command.
- Execute the command and follow the instructions on the screen.

change_mgmt_private_net

Description Changes the cluster internal communication network IP address

Syntax
```
vsx_util change_private_net
```

Input
- VSX cluster object name
- New cluster private network: New IP address for the cluster private network

Description Changes the cluster internal communication network IP address

Comments

- We recommend that you back up the management database before using this command.
- The private network IP address must be unique and not used anywhere behind the VSX gateway, cluster or Virtual Systems.
- The new cluster private network must conform to the net mask 255.255.252.0.
- Execute the command and follow the instructions on the screen.

fw fetch

Description Fetches the Inspection Code from the specified host and installs it to the kernel.

Usage `fw fetch [-n] [-f <filename>] [-c] [-i] master1 [master2] ...`

Syntax

Argument	Description
`-n`	Fetch the Security Policy from the Security Management Server to the local `state` directory, and install the Policy only if the fetched Policy is different from the Policy already installed.
`-f <filename>`	Fetch the Security Policy from the Security Management Server listed in <filename>. If filename is not specified, the list in `conf/masters` is used.
`-c`	Cluster mode, get policy from one of the cluster members, from the Check Point High Availability (CPHA) kernel list.
`-i`	Ignore SIC information (for example, SIC name) in the database and use the information in `conf/masters`. This option is used when a Security Policy is fetched for the first time by a DAIP gateway from a Security Management Server with a changed SIC name.
`master1`	Runs the command on the designated master. The name of the Security Management Server from which to fetch the Policy. You may specify a list of one or more Security Management Servers, such as `master1 master2` which will be searched in the order listed. If `targets` is not specified, or inaccessible, the policy is fetched from `localhost`.
-vs <VSID>	Fetches a Security Policy to the specified Virtual System.

change_interfaces

Description Automatically replaces designated existing interfaces with new interfaces on all virtual devices to which the existing interfaces connect.

This command is useful when converting a deployment to use Link Aggregation, especially where VLANs connect to many virtual devices.

Syntax `vsx_util change_interfaces`

Comments

- This command is interactive. Follow the instructions on the screen.
- This command supports the resume feature.
- You can use this command to migrate a VSX deployment ("Migrating from an Open Server to a VSX-1 Appliance" on page 177) from an Open Server to a Check Point appliance by using the **Management Only** mode.
- Refer to the notes ("Notes" on page 198) for additional information.

Important - You must close SmartDashboard for all Multi-Domain Security Management Domain Management Servers using the affected interfaces prior to running this command.

Using vsx_util change_interfaces

To change interfaces:

1. Close SmartDashboard for the Security Management Server and/or Multi-Domain Security Management Domain Management Servers.
2. On the management server, enter the Expert Mode and run the *vsx_util change_interfaces* command.
3. Enter the Security Management Server or Multi-Domain Security Management main Domain Management Server IP address.
4. Enter the administrator name and password as requested.
5. Enter the VSX cluster object name.
6. When prompted, select one of the following options:
   - **Apply changes to management and Security Gateway/cluster members**: Changes the on the management server, the VSX Security Gateway and cluster members.
   - **Apply changes to management Only**: Changes interface on the management server only. You must use the `vsx_util reconfigure command ("reconfigure" on page 201)` to push the updated configuration to VSX gateways or cluster members.
7. When prompted, select the interface to be replaced.
8. When prompted, select the replacement interface.
   a) You can optionally add a new interface by selecting "**Enter new interface name**". This interface must physically exist on the VSX Gateway or cluster members or the operation will fail.
   b) At the prompt, enter the new interface name. If the new interface is a bond, the interface name must match the bond name exactly (bond names are case sensitive).
9. To replace additional interfaces, enter "**y**" when prompted and repeat steps 6 through 8.
10. To complete the process, enter "**n**".
11. If you selected the **Apply changes to management only** option, run the `vsx_util reconfigure command ("reconfigure" on page 201)` to push the updated configuration to the VSX gateways or cluster members.
12. Reboot the VSX gateway and/or cluster members as appropriate.

Notes

- The **Apply changes to management and gateway/cluster members** option verifies connectivity between the management server and the VSX gateway or cluster members. In the event of a connectivity failure one of the following actions occur:
  a) If **all** of the newly changed interfaces fail to establish connectivity, the process terminates unsuccessfully.
  b) If one or more interfaces successfully establish connectivity, while one or more other interfaces fail, you may optionally continue the process. In this case those interfaces for which connectivity was established successfully will be changed. For those interfaces that failed, you must then resolve the

issue and then run the vsx_util reconfigure ("reconfigure" on page 201) command to complete the process.

- If you select the **Apply changes to management Only** option, you can select another interface from list (if any are available) or select the option to add a new interface.

change_mgmt_subnet

Description Change the gateway or member management subnet

Syntax
```
vsx_util change_mgmt_subnet
```

Input
- VSX gateway/member object name
- New subnet mask

Comments
- Backup the management database before using this command
- Only automatically generated routes are changed by the command script. You must remove and/or change all manually created routes using the previous management subnet.

To perform this action, execute the command and follow the instructions on the screen. Reboot the VSX gateway or cluster members after the command script finishes.

convert_cluster

Description Converts the cluster type from high availability to VSLS or from VSLS to high availability

Syntax
```
vsx_util convert_cluster
```

Input
- VSX cluster object name
- ClusterXL mode: **HA** for high availability or **LS** for Virtual System Load Sharing

Comments
- Backup the management database before using this command.
- To perform this action, execute the command and follow the instructions on the screen.
- When switching to high availability, all Virtual Systems are active on the same member by default. Peer Virtual Systems are standby on other members.
- When converting to VSLS, all members must be in the Per Virtual System state.

reconfigure

Description Restores a VSX configuration to a newly installed gateway or cluster member

Syntax
```
vsx_util reconfigure
```

Input
- VSX cluster member name
- SIC activation key assigned to the Security Management Server or Domain Management Server
- Retype to confirm the SIC activation key

Description Restores a VSX configuration to a newly installed gateway or cluster member

Comments

- This command is also useful for restoring a gateway or cluster member after a system failure.
- Execute the command and follow the instructions on the screen.
- A new gateway or cluster member must have the same hardware specifications and configuration as its replacement and other cluster members. Most importantly, it must have the same number of interfaces (or more) and the same management IP address.
- The new or replacement machine must be a new installation. You cannot use a machine with a previous VSX configuration.

remove_member

Description Removes a member from an existing cluster

Syntax

```
vsx_util remove_member
```

Comments

- Backup the management database before using this command
- Make certain that you remove member license before executing this command
- Execute the command and follow the instructions on the screen

show_interfaces

Description Displays selected interface information in a VSX deployment. Provides information regarding interface types, connections to virtual devices, and IP addresses. The output appears on the screen and is also saved to the *interfacesconfig.csv* file.

Syntax

```
vsx_util show_interfaces
```

Parameters

Option	Description
1) All Interfaces	Show all interfaces (physical and Warp)
2) All Physical Interfaces	Show Physical interfaces only
3) All Warp Interfaces	Show Warp interfaces only
4) A Specific Interface	Enter the interface name when prompted to a specific interface.

Note - You cannot specify a VLAN tag as a parameter for the **Specific Interface** option. You can, however, specify an interface used as a VLAN (without the tag suffix) to view all tags associated with that interface. This is illustrated in the sample output below.

Sample Output

Description Displays selected interface information in a VSX deployment. Provides information regarding interface types, connections to virtual devices, and IP addresses. The output appears on the screen and is also saved to the *interfacesconfig.csv* file.

```
Which interface would you like to display?
  1)    All Interfaces
  2)    All Physical Interfaces
  3)    All Warp Interfaces
  4)    A Specific Interface

Enter your choice: 4

Enter Interface Name:eth1

+------------------+--------------------+----+--------------+---------------+
|Type & Interface  | Virtual Device Name |VSID|  IP Address  |    Netmask    |
+------------------+--------------------+----+--------------+---------------+
|V eth1.11         |vs1                 |2   |10.1.1.11     |255.255.255.0  |
+------------------+--------------------+----+--------------+---------------+
|V eth1.44         |vs4                 |6   |10.4.4.44     |255.255.255.0  |
+------------------+--------------------+----+--------------+---------------+
|V eth1.22         |vs2                 |3   |10.2.2.22     |255.255.255.0  |
+------------------+--------------------+----+--------------+---------------+
|V eth1.33         |vs3                 |4   |10.3.3.33     |255.255.255.0  |
+------------------+--------------------+----+--------------+---------------+
|V eth1.55         |vs5                 |5   |10.5.5.55     |255.255.255.0  |
+------------------+--------------------+----+--------------+---------------+

Type:   M - Management Interface       S - Synchronization Interface
        V - VLAN Interface             W - Warp Interface
        U - Used Interface             A - Available Interface
        X - Unknown Interface          E - Error in Interface Properties
```

upgrade

Description Upgrades gateways and/or cluster members to newer versions

Syntax
```
vsx_util upgrade
```

Comments
- This command updates all VSX objects in the management database to the designated newer version.
- Backs up the management server.
- Execute the command and follow the instructions on the screen.
- After the command script finishes, execute the vsx_util reconfigure command.

view_vs_conf

Description Displays virtual device configuration and status, including troubleshooting information. This command also compares the management server database with the actual VSX gateways and cluster member configurations.

Syntax
```
vsx_util view_vs_conf
```

Output

```
Interfaces configuration table:
+-------------------------+-------------+-----------------------------
+
|Interfaces               |Mgmt |VSX GW  |
|
+----------+--------------+-------------+---------+---------+---------
+
|Name      |Address       |Mask          | Member1 | Member2 | Member3
|
+----------+--------------+-------------+---------+---------+---------
+
|eth1.51   |20.10.10.1    |255.255.255.0 |    V    |    V    |    V
|
|wrp128    |172.23.50.181 |255.255.255.0 |    V    |    V    |    V
|
+----------+--------------+-------------+---------+---------+---------
+

Interfaces Table Legend:

V: Interface exists on the gateway and matches management information
-      - Interface does not exist on the gateway.
N/A   - Fetching Virtual Device configuration from the gateway failed.
!IP  - Interface exists on the gateway, but there is an IP address
mismatch.
!MASK - Interface exists on the gateway, but there is a net mask
mismatch.
Routing table:
+-------------------------------------------------------+----+------------
-+
|Routes                                                 |Mgmt|VSX GW(s)
|
+------------+--------------+-------------+----------+----+------+-----
-+
|Destination |Mask          |Gateway      |Interface |
|Member|Member|
+------------+--------------+-------------+----------+----+------+-----
-+
|172.23.50.0 |255.255.255.0 |             |wrp128    |  V |  V   |  V
|
|20.10.10.0  |255.255.255.0 |             |eth1.51   |  V |  V   |  V
|
|20.30.30.0  |255.255.255.0 |172.23.50.82 |          |  V |  V   |  V
|
+---------------+----------------+----------------+-------------+-----+--------+---------+
```

```
+-------------------------------------------------------+-----+--------
-+
|Routes                                                 |Mgmt |VSX
GW(s)|
+---------------+-------------+------------+----------+-----+--------
-+
|Destination    |Mask         |Gateway     |Interface |     |Member3
|
+---------------+-------------+------------+----------+-----+--------
-+
|172.23.50.0    |255.255.255.0 |            |wrp128    |  V  |    V
|
|20.10.10.0     |255.255.255.0 |            |eth1.51   |  V  |    V
|
|20.30.30.0     |255.255.255.0 |172.23.50.82 |          |  V  |    V
|
+---------------+-------------+------------+----------+-----+--------
-+
Routing Table Legend:
V   - Route exists on the gateway and matches management information.
-   - Route does not exist on the gateway.
N/A  - Fetching Virtual Device configuration from the gateway failed.
!NH  - Route exists on the gateway, but there is a Next Hop mismatch.

Description Display VSLS load sharing configuration and status
Usage vsx_util vsls
Output
Sample output:
vsx_util vsls
Enter SmartCenter Server/main Domain Management Server IP address (Hit
'ENTER' for 'localhost'):
Enter Administrator Name:
Enter Administrator Password:
Enter ClusterXL Load Sharing cluster object name:

VSID| VS name | m5        | m6        | m7        | Weight |
----+---------+-----------+-----------+-----------+--------+
 10 | vs1     | 0         | 1         | 2         | 10     |
 11 | vs2     | 2         | 0         | 1         | 10     |
 12 | vs3     | 1         | 2         | 0         | 10     |
 13 | vs4     | 0         | 2         | 1         | 10     |
 15 | vs5     | 1         | 0         | 2         | 10     |
 20 | vs6     | 2         | 1         | 0         | 10     |
----+---------+-----------+-----------+-----------+--------+
 Total weight | 20        | 20        | 20        | 60     |
----+---------+-----------+-----------+-----------+--------+
Legend:
0 - Highest priority
1 - Next priority
2 - Lowest priority
```

vsls

Description Displays the Virtual System Load Sharing Menu, which allows you to perform a variety of configuration tasks for Load Sharing deployments. You perform configuration tasks ("Configuring Virtual System Load Sharing" on page 113) interactively by following the instructions on the screen.

Syntax

```
vsx_util vsls
```

Description Displays the Virtual System Load Sharing Menu, which allows you to perform a variety of configuration tasks for Load Sharing deployments. You perform configuration tasks ("Configuring Virtual System Load Sharing" on page 113) interactively by following the instructions on the screen.

Output

```
VS Load Sharing - Menu
______________________________
1. Display current VS Load sharing configuration
2. Distribute all Virtual Systems so that each
cluster member is equally loaded
3. Set all VSs active on one member
4. Manually set priority and weight
5. Import configuration from a file
6. Export configuration to a file
7. Exit

Enter redistribution option (1-7) [1]
```

Comments

- This command is interactive. Select the desired menu option and follow the instructions on the screen.

You use the *vsx_util vsls* command to perform various Virtual System Load Sharing configuration tasks, including:

1. Displaying the current VSLS configuration
2. Distributing Virtual Systems equally amongst cluster members
3. Set all Virtual Systems as active on one member
4. Manually define the priority and weight for individual Virtual Systems
5. Import VSLS configurations from comma separated value (CSV) text files
6. Export VSLS configurations to comma separated value (CSV) text files
7. Exporting and Import VSLS configurations from/to comma separated value (CSV) text files

To work with the vsx_util vsls command:

1. Run *vsx_util vsls f*rom the Expert mode on the management server
2. Select the desired choice from the **VSLS** menu

The cphaprob Command

You use the `cphaprob` command to verify cluster functionality and to debug cluster related problems. This section provides a brief overview of the `cphaprob` command and its command options. For complete documentation and use cases, refer to the ClusterXL Administration Guide (http://supportcontent.checkpoint.com/documentation_download?ID=7240).

A critical device is a process running on a cluster member that enables the member to notify other cluster members that it can no longer function as a member. The device reports to the ClusterXL mechanism regarding its current state or it may fail to report, in which case ClusterXL decides that a failure has occurred and another cluster member takes over. When a critical device (also known as a Problem Notification, or pnote) fails, the cluster member is considered to have failed.

There are a number of built-in critical devices, and the administrator can define additional critical devices. The default critical devices are:

Cluster interfaces on the cluster members.

Synchronization — full synchronization completed successfully.

Filter — the Security Policy, and whether it is loaded.

fwd — the VPN-1 daemon.

You can include these commands in scripts for automatic execution.

To produce a usage printout for `cphaprob` that shows all the available commands, type `cphaprob` at the command line and press **Enter**. The following output appears:

```
cphaprob state
cphaprob [-a] [-vs vsid] if

The following commands are NOT applicable for 3rd party:

cphaprob -d <device> -t <timeout(sec)> -s <ok|init|problem>
[-p] register
cphaprob -f <file> register
cphaprob -d <device> [-p] unregister
cphaprob -a unregister
cphaprob -d <device> -s <ok|init|problem> report
cphaprob [-i[a]] [-e] [-vs vsid] list
cphaprob -vs <vsid> register
cphaprob -vs <vsid> unregister
cphaprob igmp ................... IGMP membership status
cphaprob [-reset] [-a] ldstat ... Sync serialization
statistics
cphaprob [-reset] [-a] syncstat . Sync transport layer
statistics
cphaprob fcustat ................ Full connectivity upgrade
statistics
cphaprob tablestat .............. Cluster tables
```

The following table describes the available command options:

Argument	Description
[-vs vsid] stat	View the status all cluster members or for a specific Virtual System. *-vs* is relevant only for VSLS.
[-a] [-vs vsid] if	View the state of the cluster member interfaces and the virtual cluster interfaces. *-vs* is relevant only for VSLS.
[-i[a]] [-e] list	View the list of critical devices on a cluster member, and of all the other machines in the cluster.
-d <device> -t <timeout (sec) > -s <ok | init | problem> [-p] register	Register <device> as a critical process, and add it to the list of devices that must be running for the cluster member to be considered active.
-f <file> register	Register all the user defined critical devices listed in <file>.
-d <device> [-p] unregister	Unregister a user defined <device> as a critical process. This means that this device is no longer considered critical.
-a unregister	Unregister all user defined devices
-d <device> -s <ok | init | problem > report	Report the status of a user defined critical device to ClusterXL.
[-reset] ldstat -vs	View sync serialization statistics. *-vs* is relevant only for VSLS.
[-reset] syncstat -vs	View sync transport layer statistics. *-vs* is relevant only for VSLS.
tablestat	Displays interfaces and IP addresses for each cluster member

Index

A

Active/Standby Bridge Mode • 13, 90, 121, 169
add_member • 197
add_member_reconf • 198
Adding a Domain Management Server • 80
Adding a New Interface • 60
Adding a New Member • 107
Adding Members • 95
Adding Virtual Routers and Switches to a Domain Management Server • 81
Adding Virtual Switches • 51
Adding Virtual System to a Domain Management Server • 81
Additional Considerations for Virtual Switch Route Propagation • 29
Advanced Clustering Configuration • 122
Advanced Pages • 106
Architecture • 154
Assigning Permissions to Administrators • 78
Assigning Priorities to Virtual Systems • 152
Authentication • 106

B

Bond Failover • 132
Bond Interface & Interface Limitations • 133
Bridge Mode • 90
Bundle License Packs are Cumulative • 72

C

Calculating topology automatically based on routing information • 40, 104
Cannot Establish SIC Trust for Gateway or Cluster • 181
Capacity Optimization • 106
CCP Initiated Failover • 133
change_interfaces • 198, 200
change_mgmt_ip • 199
change_mgmt_private_net • 199
change_mgmt_subnet • 202
Changing a Cluster's MAC Source Address • 123
Changing an Existing Interface to a Bond • 148
Changing the Bond Interface Mode • 145
Changing the Cluster Management IP and/or Subnet • 106
Changing the Cluster Type • 110
Changing the Internal Communication Network IP • 107
Check Point Password • 64
Check Point Products • 38
Checking Dynamic Routing Status • 59
chpaconf failover_bond • 193
Class of Service Definitions • 155
Client/Session Authentication • 66
Cluster Deployments • 169
Cluster Management • 97
Cluster Members • 100
Clusters on the Same Layer-2 Segment • 122
ClusterXL • 101
Command Line Reference • 184
Completing the Definition • 47
Completing the VSX Wizard • 37
Completing the Wizard • 98
Components of the System • 124
Conceptual Overview • 86
Configuration Examples • 158
Configuration Guidelines • 158
Configuration Overview • 93
Configuring a Cluster for PVST+ Load Sharing • 120
Configuring Anti-Spoofing • 62
Configuring Authentication for Specific Virtual Systems • 68
Configuring Authentication for the VSX Gateway • 67
Configuring Automatic Updates • 128
Configuring Cisco Switches for Load Sharing • 145
Configuring Client/Session Authentication • 67
Configuring Cluster Members • 97
Configuring Clusters for Active/Standby Bridge Mode • 121
Configuring Clusters for STP Bridge Mode • 120
Configuring Connection Properties - General • 60
Configuring Connections Leading to Virtual Routers • 61
Configuring Dynamic Routing • 59
Configuring Interface Topology • 62
Configuring Link Aggregation for High Availability • 134
Configuring Multicast Restrictions • 62
Configuring NAT • 69
Configuring New Cluster Members • 112, 113
Configuring Password Bypass • 129
Configuring PVST+ Load Sharing • 120
Configuring RADIUS or TACACS/TACACS+ • 64
Configuring SecurID ACE/Server • 65
Configuring the Cluster Security Policy • 98
Configuring the Gateway Security Policy • 36
Configuring URL Filtering • 127
Configuring Virtual System Load Sharing • 113
Configuring Virtual Systems for Active/Standby Bridge Mode • 121
Configuring Virtual Systems for the STP Bridge Mode • 120
Configuring Virtual Systems in Bridge Mode • 119
Configuring VSX • 32
Configuring VSX High Availability • 112
Connection via a Virtual Router • 27
Connection via a Virtual Switch • 26
Connectivity Delays on Switches • 148
Context Determination • 25
convert_cluster • 202
Converting from High Availability to VSLS • 111
Converting from VSLS to High Availability • 110
Converting the Cluster • 111
Cooperative Enforcement • 106
Core Network Security • 173
cphaconf show_bond • 192
cphaprob -a if • 193
cpqos class add • 156

cpqos class del • 157
cpqos class show • 157
cpqos install • 157
cpqos stats • 157
cpqos uninstall • 157
Creating a Bond in a New Deployment • 138
Creating a New Cluster • 93
Creating a New Deployment • 134
Creating a New Domain Object • 76
Creating a New Virtual Router • 55
Creating a New Virtual System • 42
Creating a New VSLS Cluster • 114
Creating a New VSX Gateway • 32
Creating Interface Bond in Load Sharing Mode • 138, 141
Creating the Cluster. • 135, 141
Creation Templates • 102
Custom Configuration or Override - Non-Bridge Mode • 45
Custom Configuration or Override in the Bridge Mode • 46

D

Data Centers • 177
Data Centers in an Enterprise • 177
Dedicated Management Interface (DMI) • 17
Defining an Additional Multi-Domain Server in the SmartDomain Manager • 75
Defining Cluster General Properties • 94
Defining Cluster Interfaces • 96
Defining Domains and Servers • 76
Defining General properties • 55
Defining General Properties • 43
Defining Multi-Domain Servers • 74
Defining Network Configuration • 43
Defining Physical Interfaces • 35
Defining Slave Interfaces as Disconnected • 134, 136, 139, 142
Defining Source-Based Routing Rules • 58
Defining the General Properties • 52
Defining the Interface Bond • 134, 136
Defining the Network Configuration • 52
Defining the Network Topology • 55
Defining the Spanning Tree Structure • 119, 120
Defining the URL Filtering Policy • 127
Defining VSX Gateway General Properties • 33
Deleting a Bond • 146
Deleting a Member • 108
Deleting a Virtual Router • 58
Deleting a Virtual Switch • 53
Deleting a Virtual System • 51
Deleting a VSX Gateway • 41
Deleting an Interface • 63
Deploying VSX • 166
Deployment Scenarios • 91
Detaching Interfaces from a Bond • 146
Differentiated Services Support • 154
Direct Connection to a Physical Interface • 25
Disabling Dynamic Routing • 59
Disabling VSLS Options • 110
Disabling VSX Resource Control Components • 151
Distributing Virtual Systems Amongst Members • 115
Distributing Virtual Systems for Equal Member Loading • 115
Domain Assigned Administrators Page • 78
Domain Assigned GUI Clients Page • 79
Domain Properties Page • 77
Dynamic Routing • 30, 173
Dynamic Routing in ClusterXL • 124

E

Enabling Active/Standby Bridge Mode for a New Member • 121
Enabling Active/Standby Bridge Mode for Existing Members • 121
Enabling and Configuring Active/Standby Bridge Mode • 121
Enabling Dynamic Routing • 59
Enabling Dynamic Routing Protocols • 124
Enabling Per Virtual System High Availability • 113
Enabling STP Bridge Mode when Creating Member • 120
Enabling the Per Virtual System State Mode • 113
Enabling the STP Bridge Mode for Existing Members • 120
Enabling URL Filtering • 127
Enabling VSLS • 113
Enabling VSX Gateway High Availability • 112
Enabling VSX Resource Control Components • 151
Enslaving Interfaces to a Bond • 145
Enterprise Deployments • 172
Establishing SIC Trust • 34
Example • 192, 193
Example RAID Monitoring OIDs • 163
Example Sensors Monitoring OIDs • 164
Exporting a VSLS configuration • 118
Exporting and Importing VSLS Configurations • 117

F

Failover Support for VLANs • 133
Failure Recovery • 90, 124
Firewall Commands • 184
For 802.3ad: • 145
For Both Shared and Private Options on SecureID Connections • 66
For More Information • 74
For XOR: • 145
Forwarding to Destination • 27
Fully Meshed Redundancy via Interface Bonding • 131
Functional Overview • 126
fw fetch • 186, 200
fw getifs • 184
fw monitor • 185
fw tab • 185

G

Gateway Cluster Member List • 100
General Properties • 99
General Troubleshooting Steps • 180
Global Objects • 78
Global Policy Page • 77

H

Hardware Health Monitoring • 13, 161
High Availability • 31
High Availability Licenses • 73
High Availability Overview • 131
High Performance Security • 12
How Link Aggregation Works • 131
How VSX Works • 11

I

Important Information • 3
Importing a VSLS configuration • 119
Inbound Prioritization • 154
Initializing SIC Trust • 34
Install Policy Error Using VSX Creation Wizard • 182
Installing a New Multi-Domain Server • 74
Interfaces • 20, 40, 103
Internal Communication Network • 84
Internal Host Cannot Ping Virtual System • 183
Internal IP Address and Net Mask • 101
Internal Network Deployment Strategies • 166
Internal Virtual Router with Source-Based Routing • 168
Introduction • 22, 86, 126, 166, 180
Introduction to Hardware Health Monitoring • 161
Introduction to VSX • 9
Introduction to VSX Clusters • 82
IPS • 78, 105

K

Key Features and Benefits • 12

L

Latency • 155
License Violations • 73
Licensing VSX with Multi-Domain Security Management • 71
Limitations • 30
Limitations of VSX/Domain Management Server Bundle Licenses • 73
Link Aggregation • 13
Link Aggregation - Load Sharing Mode • 137
Link Aggregation CLI Commands • 192
Link Aggregation Overview • 130
Link Aggregation Terminology • 130
Link State Initiated Failover • 132
Load Sharing Overview • 132
Local Management Connection • 16
Logs and Masters • 106

M

Making the Change Permanent • 123
Managed Service Providers Using Multi-Domain Security Management • 175
Management Interface • 17
Management Model Comparison • 24
Management Server Communication - SIC • 24
Management Server Connections • 15
Managing VSX Clusters • 93
Manually Assigning Priorities and Weights for a Single Virtual System. • 115
Member Failure Scenario • 88
Migrating from an Open Server to a VSX-1 Appliance • 178
Modifying a Cluster Definition • 98
Modifying a Virtual Router Definition • 56
Modifying a Virtual System Definition • 47
Modifying an Interface Definition • 63
Modifying Cluster Properties • 98
Modifying Existing Domains and Servers • 81
Modifying Virtual Switches • 52
Modifying VSX Gateway Definitions • 37
Monitoring all VLANs with ClusterXL • 123
Monitoring CPU Resource Utilization • 152
Monitoring the Highest and Lowest VLAN IDs • 133
Multi-Domain Security Management Licenses • 71
Multi-Domain Security Management Model • 23

N

NAT • 30, 105
Non-Dedicated Management Interface • 17
Non-Stop Security • 13
Normalized VSLS Deployment Scenario • 88
Notes • 68, 153, 199, 201
Notes to the Upgrade Process • 110

O

Operating System Password • 64
Optimizing VSX • 150
Organizational Deployment Strategies • 172
Overlapping IP Address Space • 29
Overview • 15, 25, 32, 70, 119, 150, 153

P

Password Bypass • 129
Per Virtual System High Availability • 85, 171
Performing Manual Updates • 129
Perimeter Security • 175
Physical Clusters • 82
Physical Interfaces • 21, 102
Physical Internal Interface for Each Virtual System • 167
Physical Network Topology • 11
Placing All Active Systems on the Same Member • 115
Planning a Cluster Deployment • 83
Policy with Global Scope • 154
Preliminary Steps • 108
Priority and Drop Precedence • 156
Priority and LLQs • 156
Private • 65, 66
Processing Options • 118
Product Names • 9

Q

QoS Configuration • 156
QoS Default Configuration • 158
QoS Enforcement • 153
QoS Features • 155
QoS Management • 155
QoS Policy File • 158

R

Radius • 64
RAID Monitoring with SNMP • 162
reconfigure • 202
Reconfiguring Interface Connections • 147
Reconfiguring the Bond using SmartDashboard • 137, 144
Reconfiguring Topology • 136, 144
Reconfiguring Topology using vsx_util change_interfaces • 137, 144
Redistributing Active Virtual Systems to One Member • 110
Re-establishing SIC Trust with Virtual Devices • 182
Remote Access • 106
Remote Management connection • 16
remove_member • 203
Removing a Bond Interface from a VSX Gateway or Cluster Member • 147
Removing a Bond Interface From a VSX Object • 147
Removing a Bond Interface from Virtual devices • 147
Removing IP Addresses from Slave Interfaces • 135, 141
Report Results • 193
Requirements • 86
Resource Allocation • 155
Resource Control System Components • 150
Route Propagation • 28
Route Propagation using a Virtual Router • 28
Route Propagation using a Virtual Switch • 28
Routes • 40, 103
Routing Between Virtual Systems • 27
Routing Overview • 27
Routing Table Synchronization • 124
Rules & Security Policies • 32

S

Sample Command Output • 111
Sample Configuration of PortFast on a Cisco Switch • 149
Sample Differentiated Services Implementation • 158
Sample Traffic Types • 158
Scalable Virtual Environment • 12
Secure Internal Communication (SIC) • 38
SecurePlatform • 13
SecurID • 64
Security Enforcement • 27
Security Gateway Deployment on a Physical Network • 166
Security Management Model • 23
Selecting and Opening an Existing Interface • 63
Selecting Creation Templates • 33, 94
Sensors Monitoring with SNMP on Power-1 and UTM-1 Appliances • 164
Sensors Monitoring with SNMP on VSX-1 Appliances • 163
Separate Interfaces in Bridge Mode • 44
Setting Affinities • 140, 143
Setting Critical Required Interfaces • 139, 142
Shared • 64, 65
Shared Interface or Separate Interfaces • 44
show_interfaces • 203
SIC Trust Problems with new Virtual Devices • 181
Source Cluster MAC Addresses • 122
Source-Based Routing • 29
Spanning Tree Protocol (STP) Bridge Mode • 90
Starting the Add Domain Wizard • 76
STP Bridge Mode • 119
Supported Authentication Schemes • 63
Supported Cluster Environments • 83
Sync Networks Do Not match • 182
Synchronization • 102
Synchronization Network • 84
Syntax • 192, 193

T

TACACS, TACACS+ • 64
Terminology • 126
The cphaprob Command • 207
The cpqos Command • 156
The Default Weight Factor • 151
The Effect of Upgrading on Authentication Processes • 66
The Trial Period • 73
The VSX Gateway • 15
The vsx_util Command • 196
Three Layer Hierarchical Model • 91
Topology • 103
Tracking Activity with SmartView Monitor • 69
Troubleshooting Bonded Interfaces • 148
Troubleshooting SIC Trust Initialization Problems • 34
Troubleshooting Specific Problems • 181
Troubleshooting Workflow • 148
Typical VSX Deployments • 13

U

Unnumbered Interface Limitations • 22
Unnumbered Interfaces • 22
Updating the Content Inspection Database • 128
upgrade • 204
Upgrading a Member to the Current Version • 109
Upgrading an Existing Deployment • 135, 141
Upgrading Cluster Members • 108
Upgrading Licenses • 73
URL Filtering • 13
URL Filtering Acceleration • 129
User Defined Weights • 151
Using Password Bypass • 129
Using the vsx_util vsls Command • 114
Using Virtual Switches in a Cluster • 92
Using VSX with Multi-Domain Security Management • 70
Using vsx_util change_interfaces • 198, 201

V

Verifying that the Bond is Functioning Properly • 135, 136, 140, 144
view_vs_conf • 204
Viewing VSLS Status • 116

Virtual Devices • 18
Virtual IP Addresses • 84
Virtual IP Integration • 124
Virtual Network Device Configuration • 35
Virtual Router - Advanced • 57
Virtual Router - General Properties • 56
Virtual Router - IPS • 57
Virtual Router - Logs and Masters • 57
Virtual Router - Topology • 57
Virtual Routers • 19
Virtual Switch - General Properties • 53
Virtual Switch - Topology • 53
Virtual Switches • 20
Virtual System • 18
Virtual System - Advanced • 51
Virtual System - Authentication • 50
Virtual System - Capacity Optimization • 50
Virtual System - General Properties • 47
Virtual System - IPS • 49
Virtual System - Logs and Masters • 50
Virtual System - NAT • 49
Virtual System - Remote Access • 50
Virtual System - Topology • 48
Virtual System - VPN • 49
Virtual System Autonomy • 18
Virtual System Failure Scenario • 89
Virtual System in Bridge Mode • 18
Virtual System Load Sharing (VSLS) • 31, 86, 171
Virtual System Priorities • 151
Virtual System Priority • 87, 116
Virtual System States • 88
Virtual System Weight • 87, 117
Virtual Systems in the Bridge Mode • 169, 170
Virtual Systems with Internal VLAN Interfaces • 167
VLAN Interfaces • 21
VLAN Shared Interface Deployment • 91
VLAN Shared Interface Deployment - Active Standby Bridge Mode • 170
VPN • 105
VPN Domain • 104
vsls • 206
VSLS Configuration File • 117
VSX Architecture and Concepts • 15
VSX Cluster Architecture • 84
VSX Clustering Overview • 82
VSX Clusters • 30, 83
VSX Command • 188
VSX Diagnostics and Troubleshooting • 180
vsx fetch • 188
vsx fetchvs • 189
VSX Gateway - Advanced Pages • 41
VSX Gateway - Authentication • 41
VSX Gateway - Capacity Optimization • 41
VSX Gateway - Creation Templates • 39
VSX Gateway - General Properties • 37
VSX Gateway - Logs and Masters • 41
VSX Gateway - NAT • 41
VSX Gateway - Physical Interfaces • 39
VSX Gateway - Remote Access • 41
VSX Gateway - Topology • 40
VSX Gateway - VPN • 41
VSX Gateway High Availability • 85
VSX Gateway Management • 36
VSX Gateway Recovery • 42
VSX Gateway/Cluster Member Licenses • 72
vsx get • 189
VSX Glossary • 10
VSX High Availability • 85
VSX Limitations • 67
VSX Management Overview • 22
VSX Overview • 10
VSX Provisioning • 74
vsx resctrl enforce • 194
vsx resctrl monitor • 194
vsx resctrl reset • 194
vsx resctrl start • 195
vsx resctrl stat • 195
vsx resctrl traffic_stat • 194
VSX Resource Control • 150
VSX Resource Control Commands • 193
VSX Routing Concepts • 27
vsx set • 189
vsx sic reset • 192
vsx start_dr • 191
vsx stat • 190
VSX Traffic Flow • 25
VSX Virtual Network Topology • 12
VSX Virtual System Deployment Strategies • 167
VSX/Domain Management Server Bundle Licenses • 72

W

Warning Regarding Use of PortFast • 148
Warp Links • 21
Where Used • 101
Working with Authentication • 63
Working with Cluster Members • 107
Working with Dynamic Routing • 59
Working with Interface Definitions • 60
Working with Link Aggregation • 130
Working with Network Address Translation • 68
Working with Source-Based Routing • 58
Working with URL Filtering • 126
Working with Virtual Devices • 81
Working with Virtual Routers • 53
Working with Virtual Switches • 51
Working with Virtual Systems • 42
Working with VSX Gateways • 32
Working with VSX Resource Control • 151
WRED • 155